Implementing an Enterprise GIS Project

Lana Tylka

Jennifer Prather

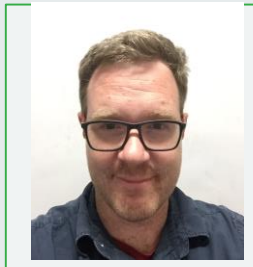
Craig Venker

July 9, 2017

Introductions

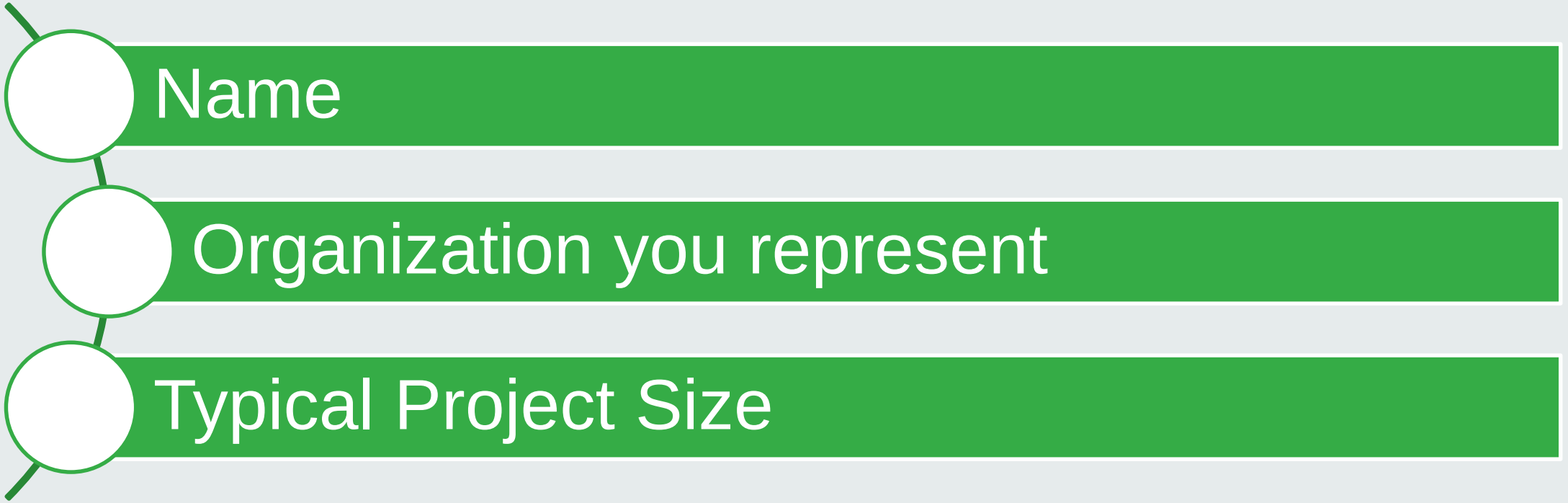
Lana Tylka

Jennifer Prather



Craig Venker

Introductions



The **purpose** of this workshop is to provide you with an **understanding** of the Enterprise GIS implementation **process** and its **components**

Agenda

8:30 – 9:00	Opening and Introductions
9:00 – 9:30	Case Study
9:30 – 10:30	Vision, Goals, and Objectives
10:30 – 10:45	Morning Break
10:45 – 11:30	Assess and Plan
11:30 – 1:00	Lunch Break
1:00 – 2:00	Building IOC
2:00 – 2:15	Afternoon Break
2:15 – 3:30	Building IOC
3:30 – 3:45	Afternoon Break
3:45 – 4:30	Building IOC
4:30 – 5:00	Evaluate, Review, and Plan Ahead

- Sign up sheet
- Class materials
- Questions
- Break and lunch times
- Bathrooms
- Evacuation procedures

Course Scenario

https://www.youtube.com/watch?v=_3MyyiuPAqE

DASHBOARD

Crop Health

Product Selection

Data Management

Ordering

Crop Scout

Suggestions

History

Yield

Analysis

Weather

Imagery

3rd Party

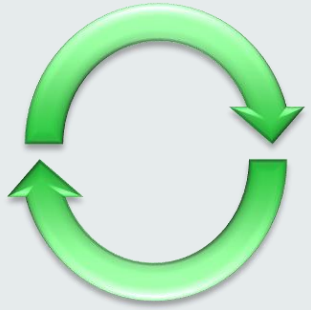
Yield Data and Weather Data

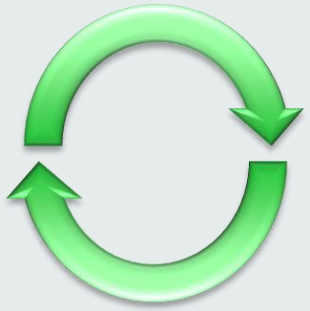
iPad App

iPhone App

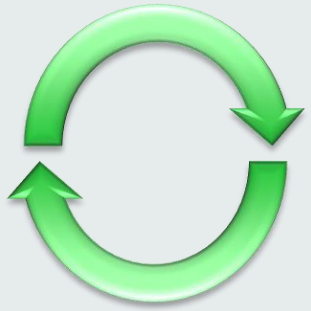
Admin Site

AGOL

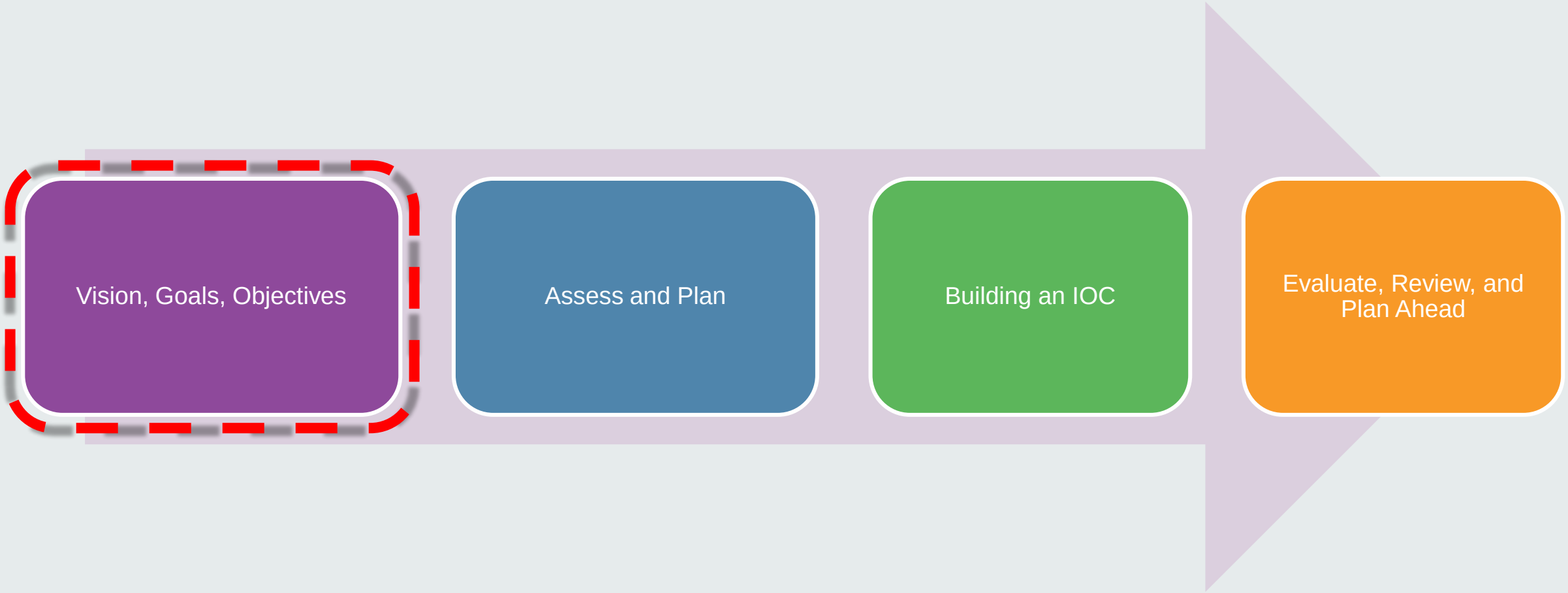

UI/UX


Performance

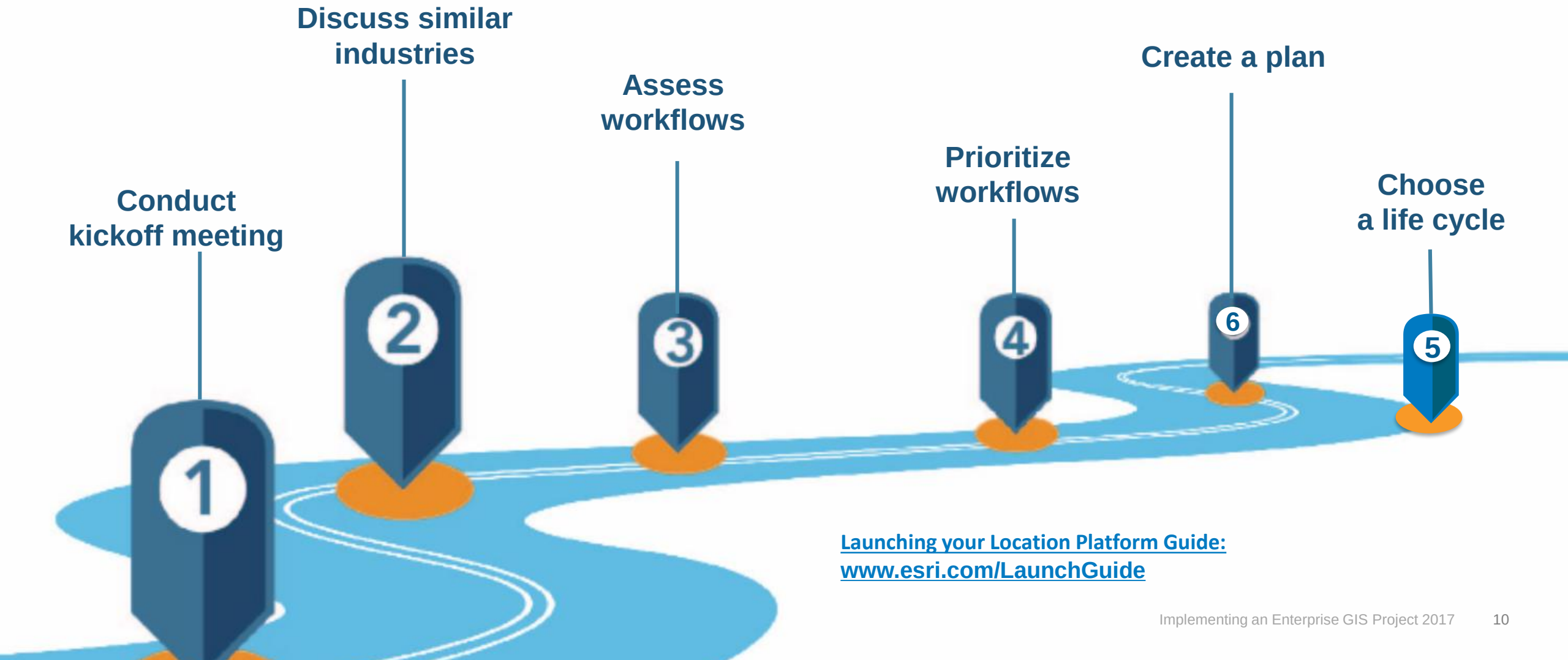
Maintenance


Management

Assess the Following Components



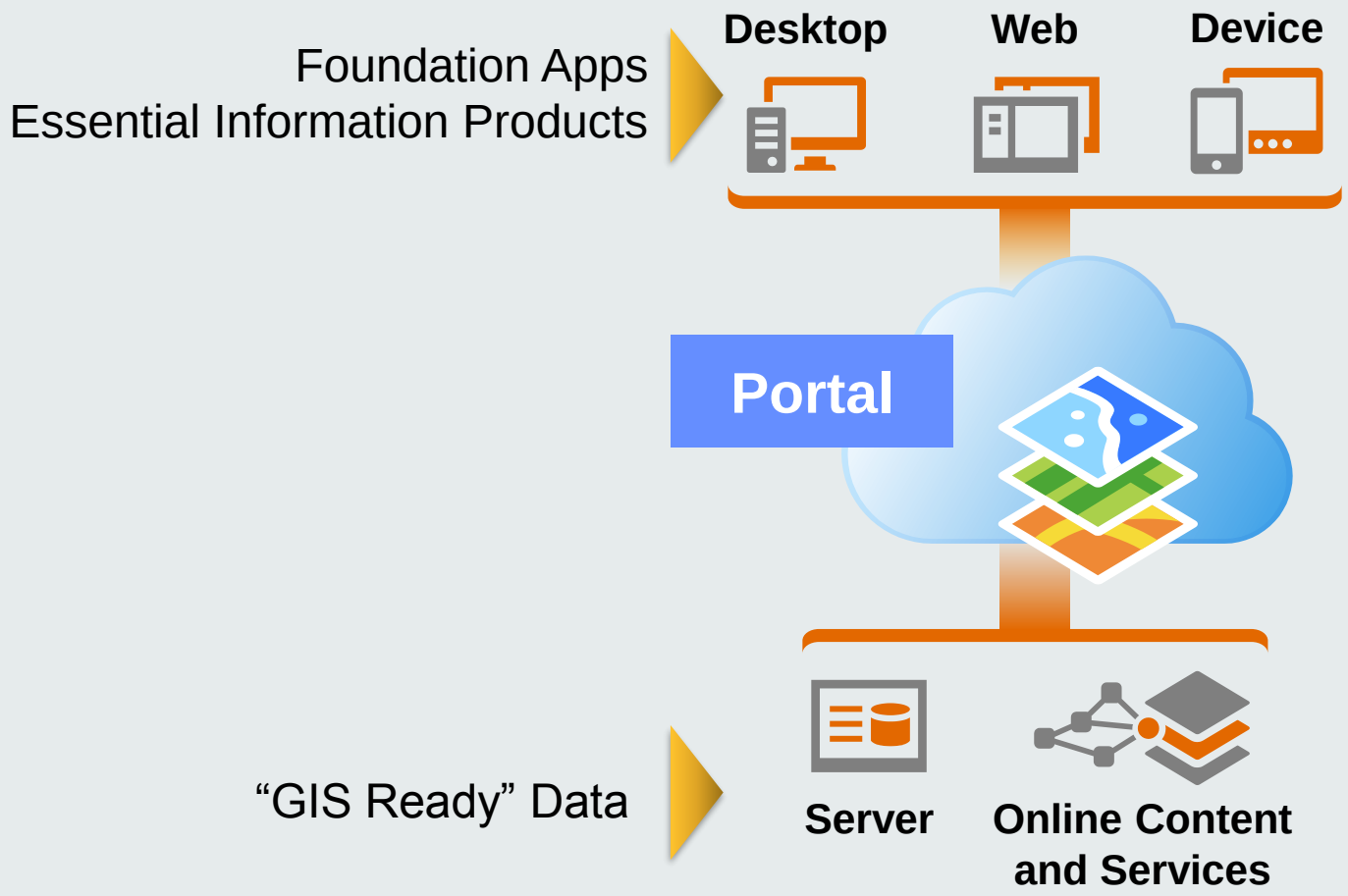
How to Start…



[Launching your Location Platform Guide: www.esri.com/LaunchGuide]

Implementing the Platform

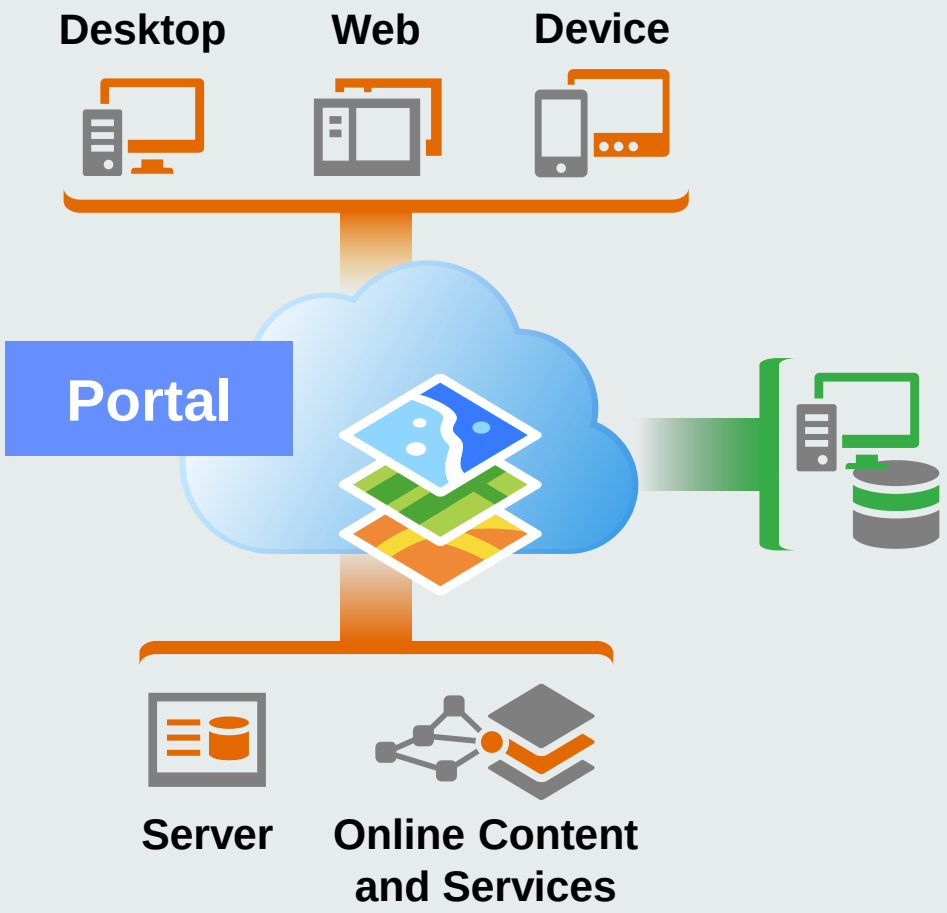
Rapid Deployment of Initial Operating Capability



…COTS configuration for quick results

Implementing the Platform

Realization of an Integrated Enterprise System



Extended / Custom Apps

Integration with business systems

Advanced integrated workflows

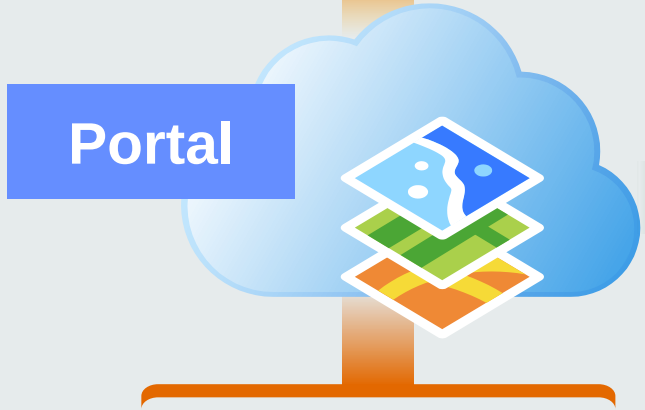
Content Production Systems
Data modernization

…Reaching across the enterprise with lasting success

Patterns of Implementation

Initial Operating Capability

Foundation Apps
Essential Information Products



“GIS Ready” Data

Integrated Enterprise System

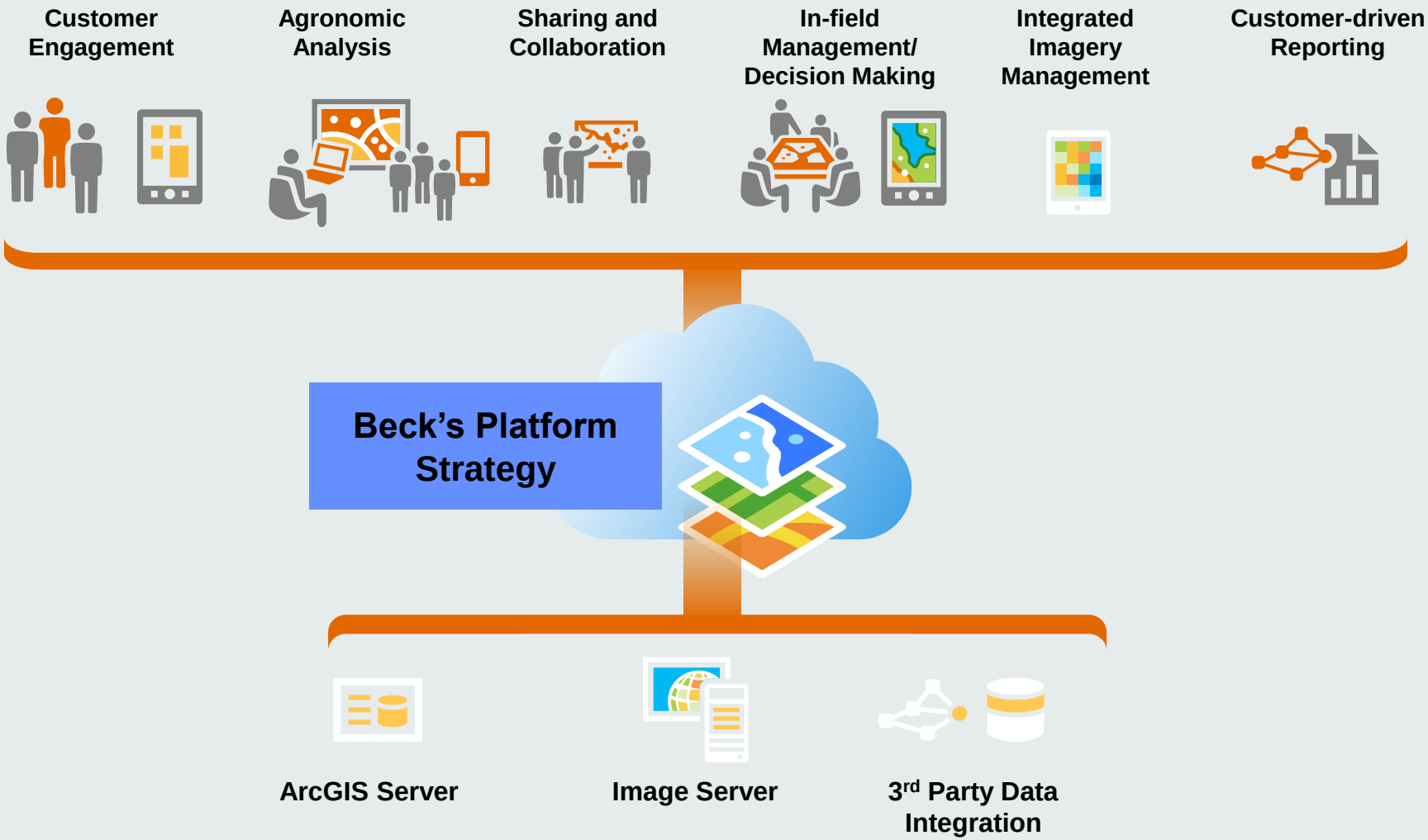
Extended / Custom Apps

Integration with
business systems

Advanced integrated
workflows

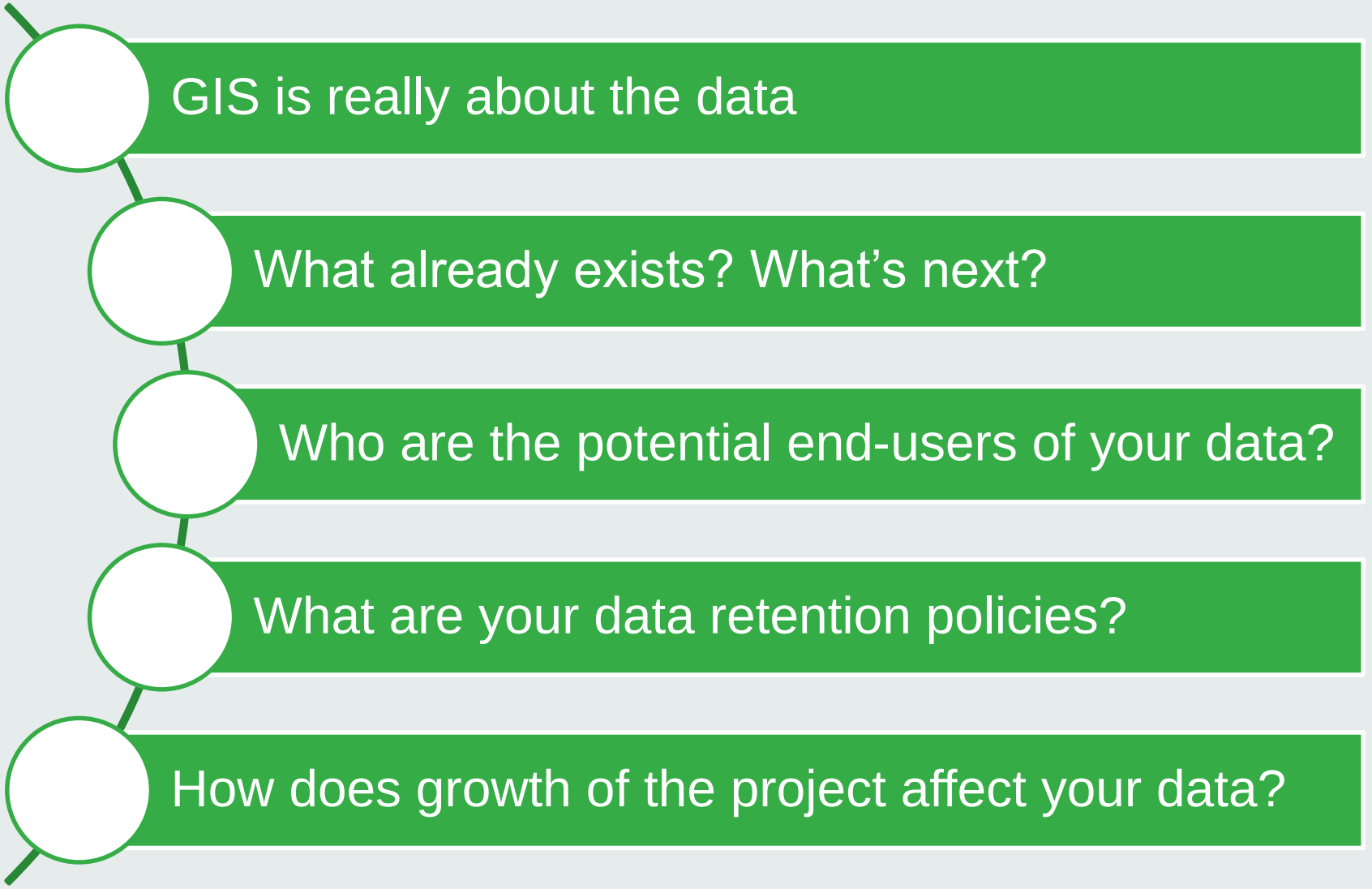
Content Production Systems
Data modernization

Beck's Platform Vision



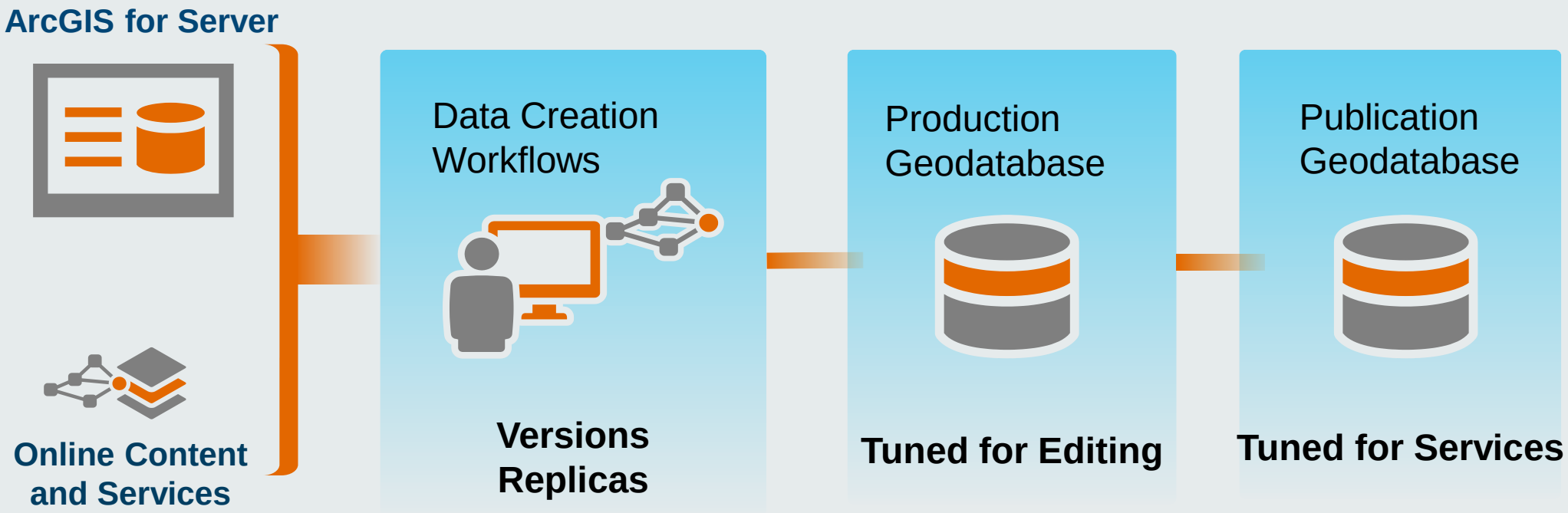
Exercise <Your Agency Name> Platform Vision

Considerations



Geodata Management Strategies

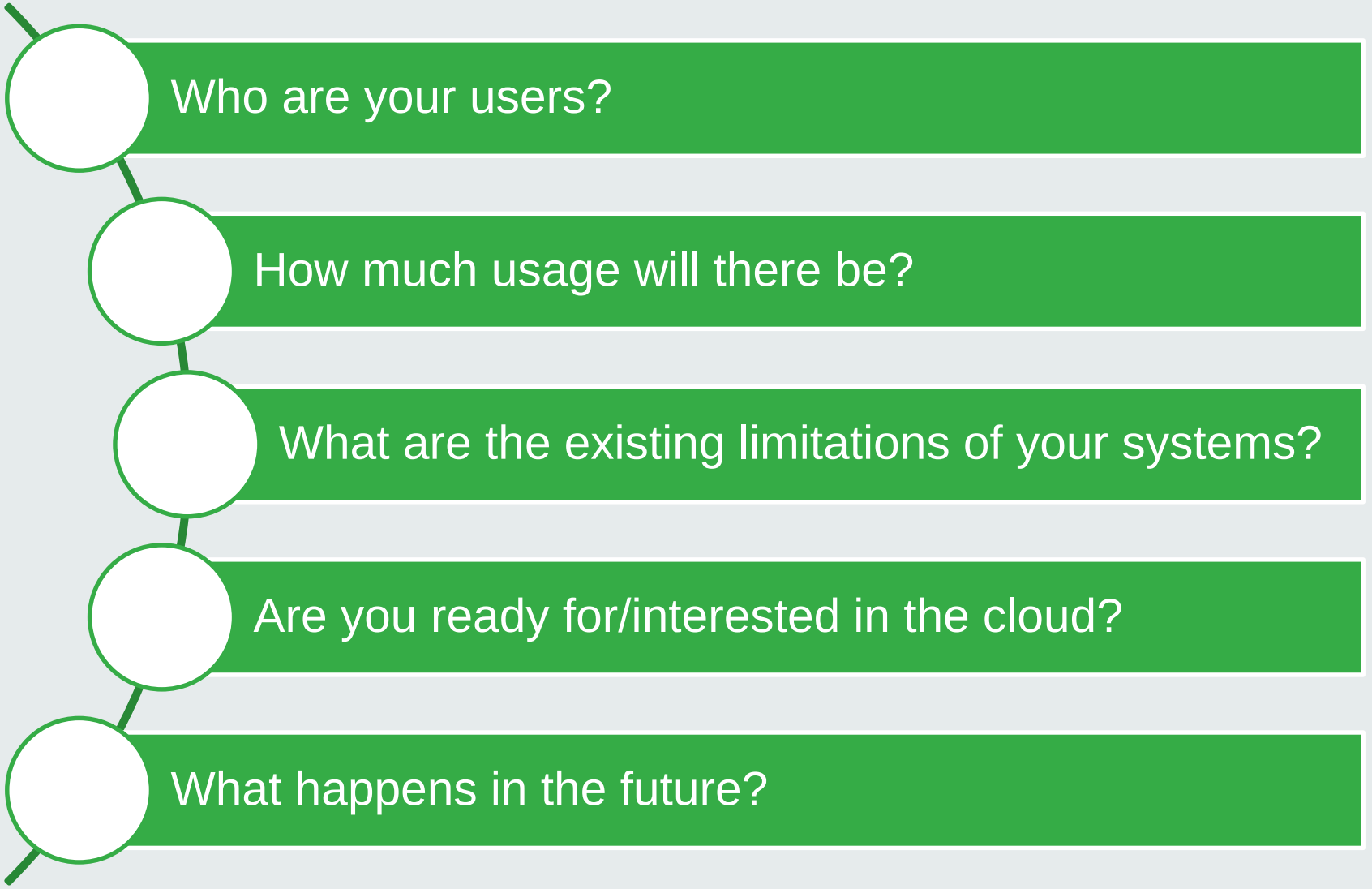
- **Separate production from publication environments!**
- **Make Data Owners responsible for (and proud of) their data.**
- **Structure data to optimize for your information products**
- **Consider your deployment environment (Web? Desktop? Device?)**



Geodata Management Best Practices

- Design and group data according to maintenance and security policies, not the application using it
- Feature datasets are for topologies and networks, not to make data look pretty in ArcCatalog
- Don’t overthink your QA and versioning to avoid paralysis by analysis
- Consider a default-open strategy
- Don’t recreate or re-host data from external providers if possible
- Host heavily-used data and services in a cloud environment to reduce performance impacts

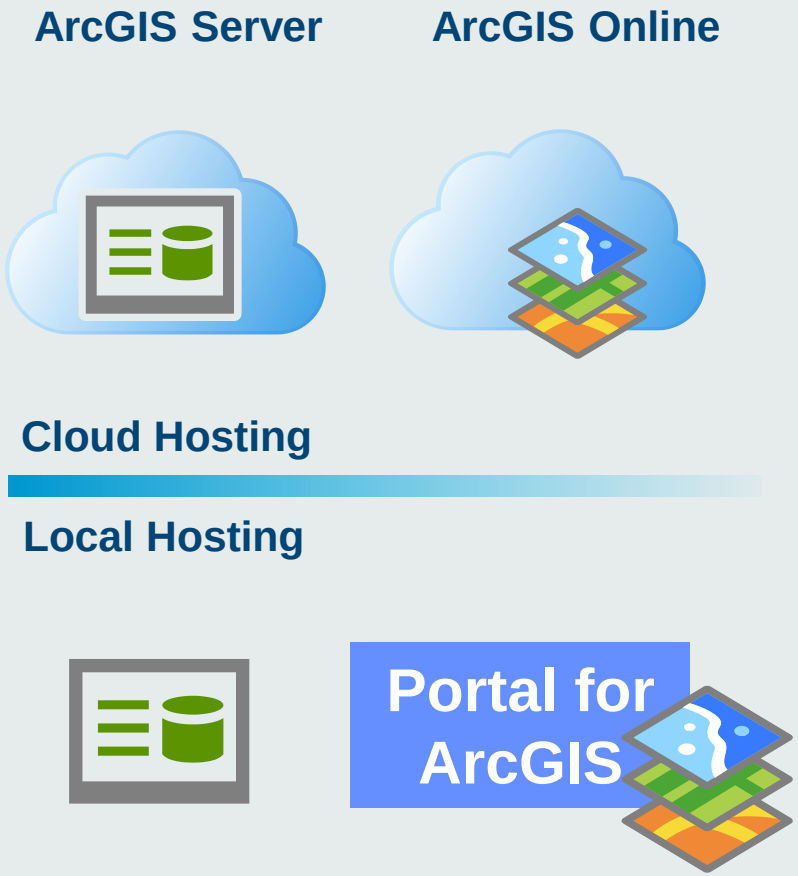
Base your Infrastructure Decisions on Facts



Key Infrastructure & Deployment Considerations

Things to Think About

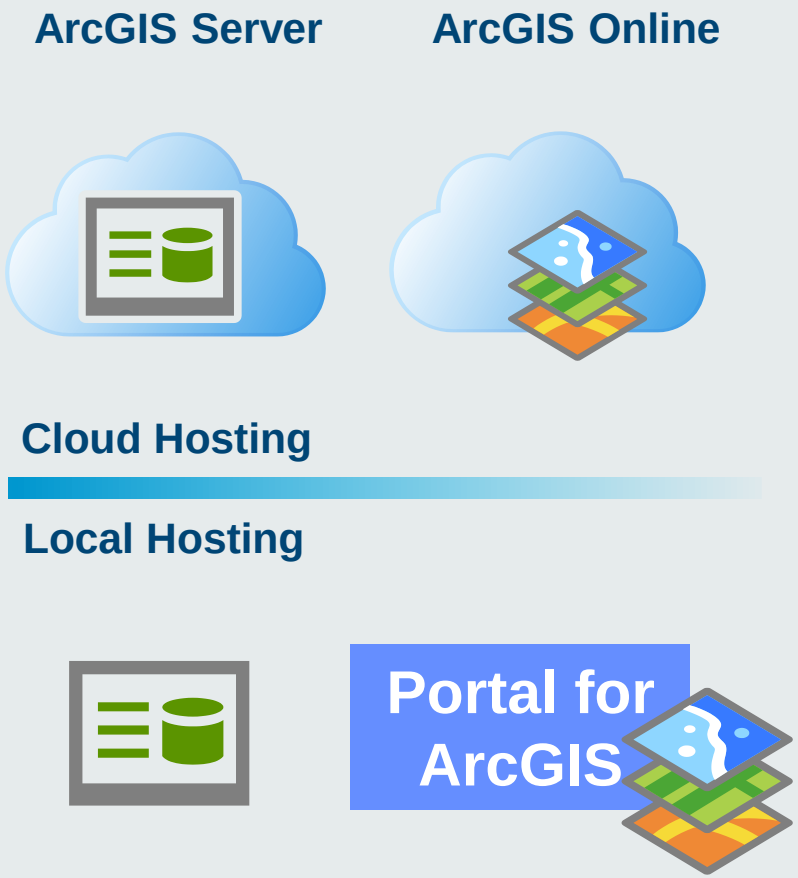
- **Cloud Hosting vs. On-Premises**
- **Security: Access, Authentication, Auditing, Standards**
- **Network Access**
- **Reliability**
- **Mobile Device Access**
- **Contractors, 3rd Party and Public**
- **Existing users, apps, infrastructure**
- **Databases and storage types**
- **Physical vs. Virtualization**



Using the Cloud vs. "Your Own" Infrastructure

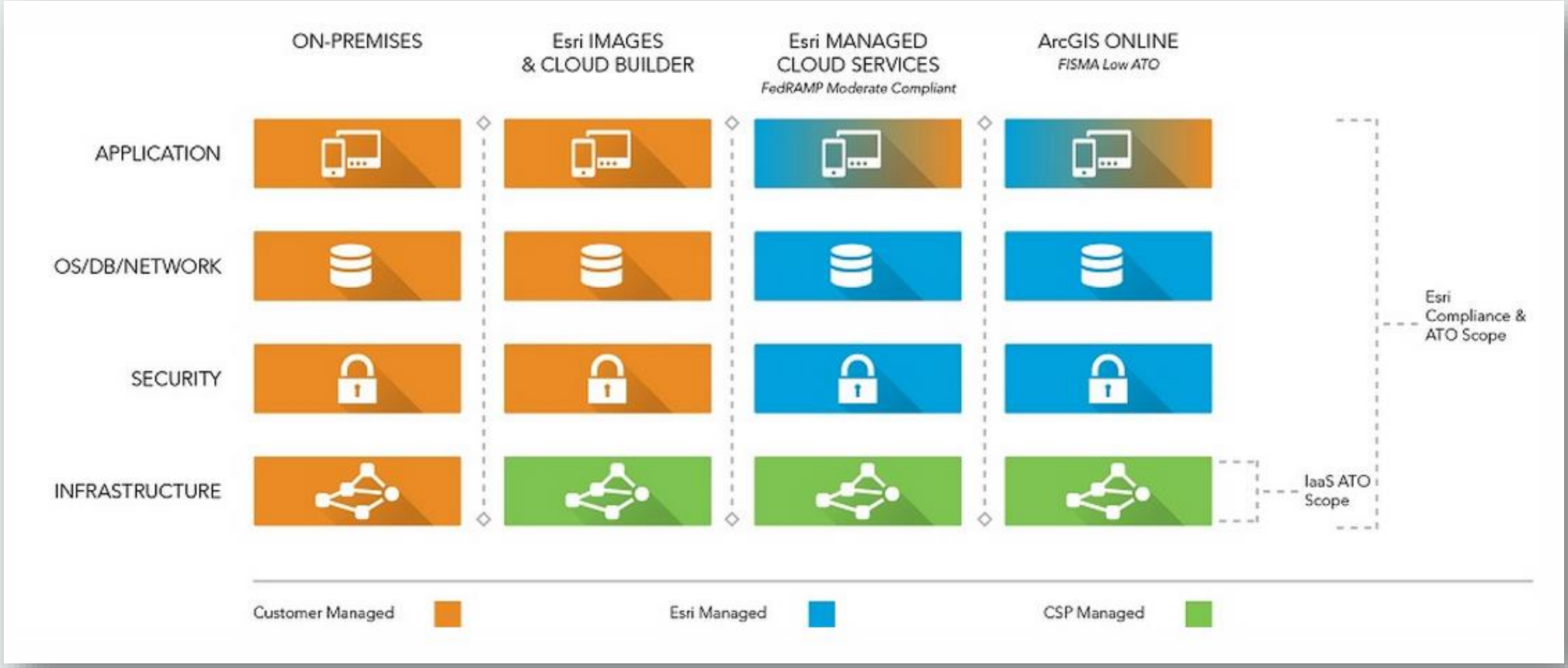
- **Cloud Deployments**
  - **Reduced capital expenditure**
  - **Reduced maintenance**
  - **Rapidly scalable**
  - **Many ready to use resources**
  - **Many deployment options**
  - **Trend is to the Cloud**

- **Managing your Own Environment**
  - **More control, but also more management**
  - **Security: Inside firewall (but the cloud is inside a firewall too)**
  - **Network: LAN is faster than WAN**
  - **Reliability: Not affected by Internet outages**



Web GIS Deployment Patterns

Cloud Deployment Models



Source: trust.arcgis.com

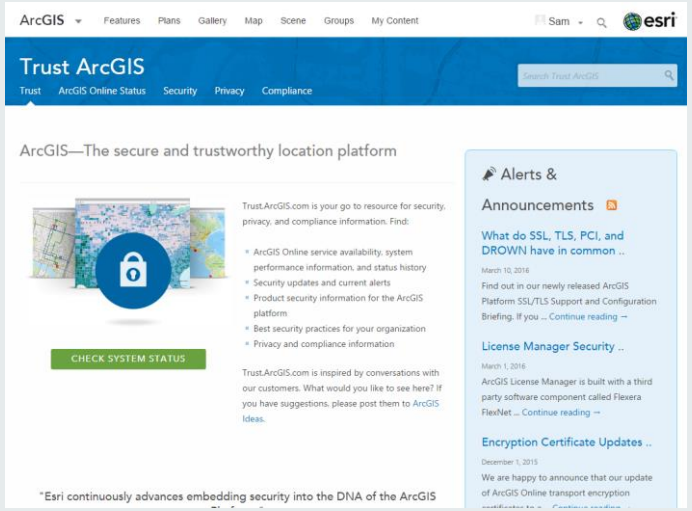
Key Considerations

- **Security**
  - **Confidence that your customer data and organizational data is safe**
- **Authentication**
  - **The GIS needs to ‘know’ who you are, and trust that you are who you say you are**
- **Authorization**
  - **Separation of roles and responsibilities**
  - **Least Privileges Model**

External

Inside your Firewall

trust.arcgis.com



Who are your Users?

- **Internal Users**
  - **Usually your first tier of support**
  - **Everything ‘works’ in the office**
- **3rd Parties**
  - **Contractors, outside consultants, neighboring organizations**
- **External users**
  - **How does the public interface with your GIS? How do they make use of your data and services (if at all)?**

External

Inside your Firewall

Data Storage and Usage

- **Where is your data now?**
  - In an enterprise geodatabase?
  - On a highly-available file share?
  - In an existing business system?
  - On someone's laptop?
  - External Hard Drives?
  - DVDs?
- **Priorities**
  - Quality
  - Availability
  - Redundancy

External

Inside your Firewall

Assess Your Current Data Flows

- **What data is needed to support the key business needs**
- **What are easy GIS data sets that can be stood up quickly or are already accessible**
- **Who is responsible for/owns which GIS layers?**
- **What are the data exchange protocols?**
- **What is the format of the GIS data?**
- **Are there any GIS data standards in your organization?**
- **What security is associated to your data?**

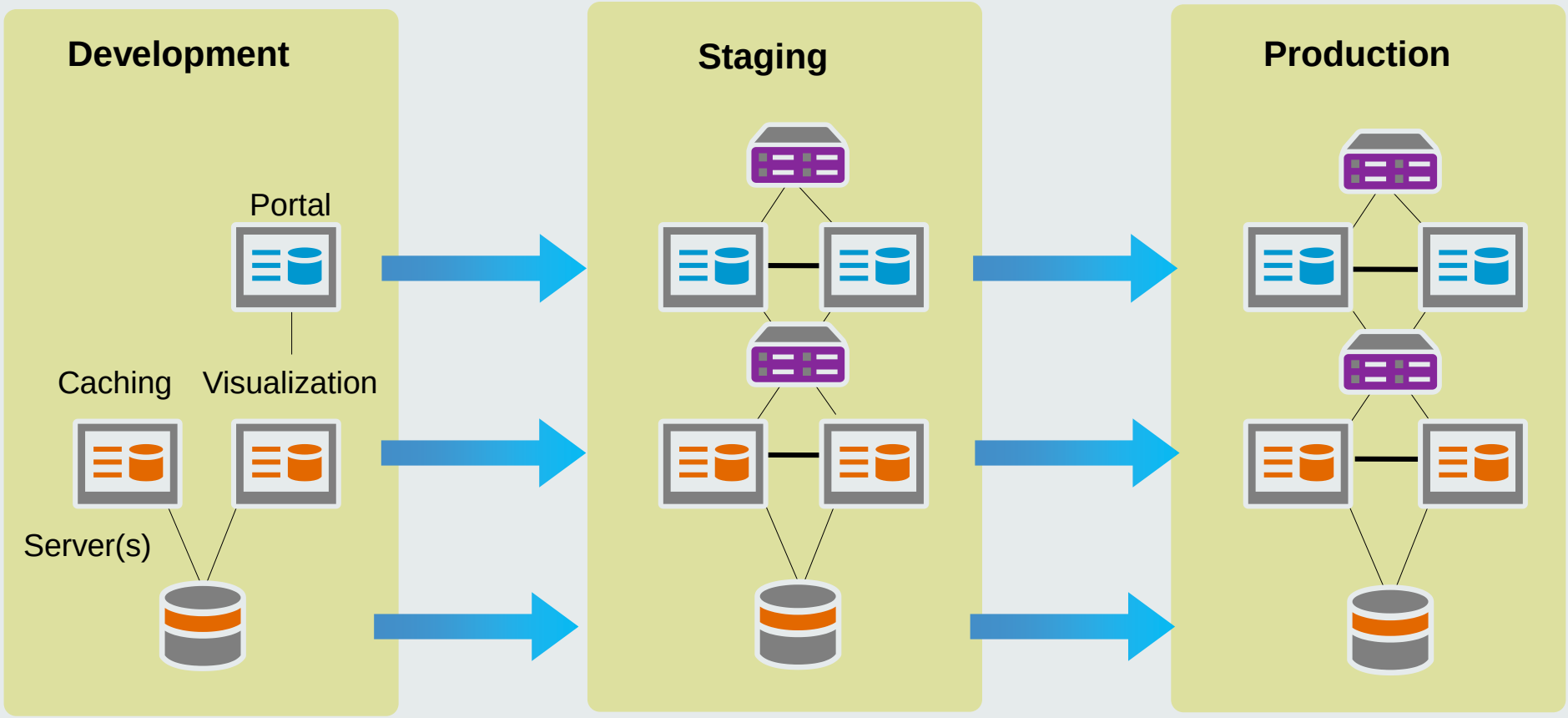
Publish Data

- Web Services and Map Services are essential to Web GIS and modern mapping workflows
- Design services around data, not around application: reusable, flexible, optimized
- Consider services the primary data access method
- Make data application-agnostic and easily understood
- Be ready to leverage outside data and services in your information products

System Architecture:

https://www.esri.com/~/media/Files/Pdfs/products/arcgis-platform/architecting-the-arcgis-platform

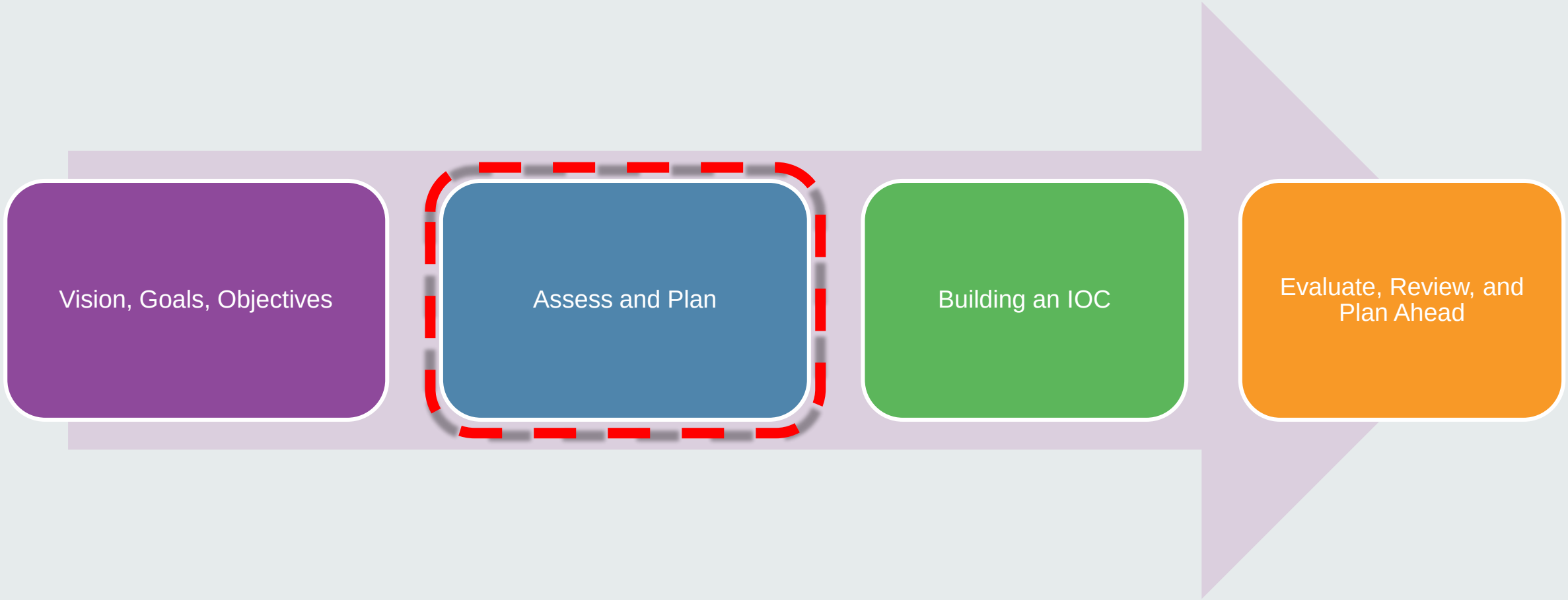
Leverage best practices: ex. Environment Isolation



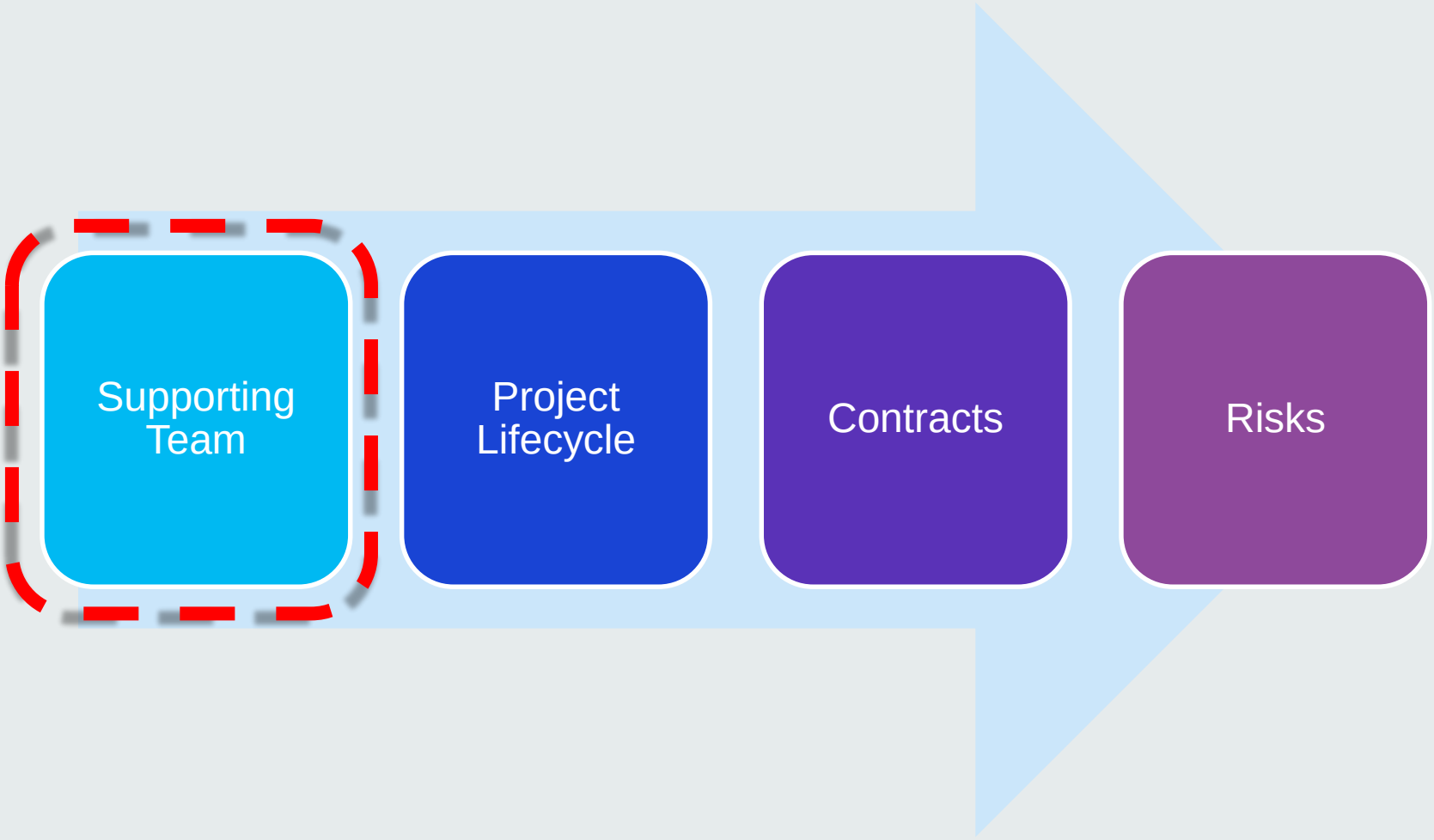
Exercise <Your Agency Name> Platform Vision

Update with what you have learned…

Assess the Following Components



Assess and Plan

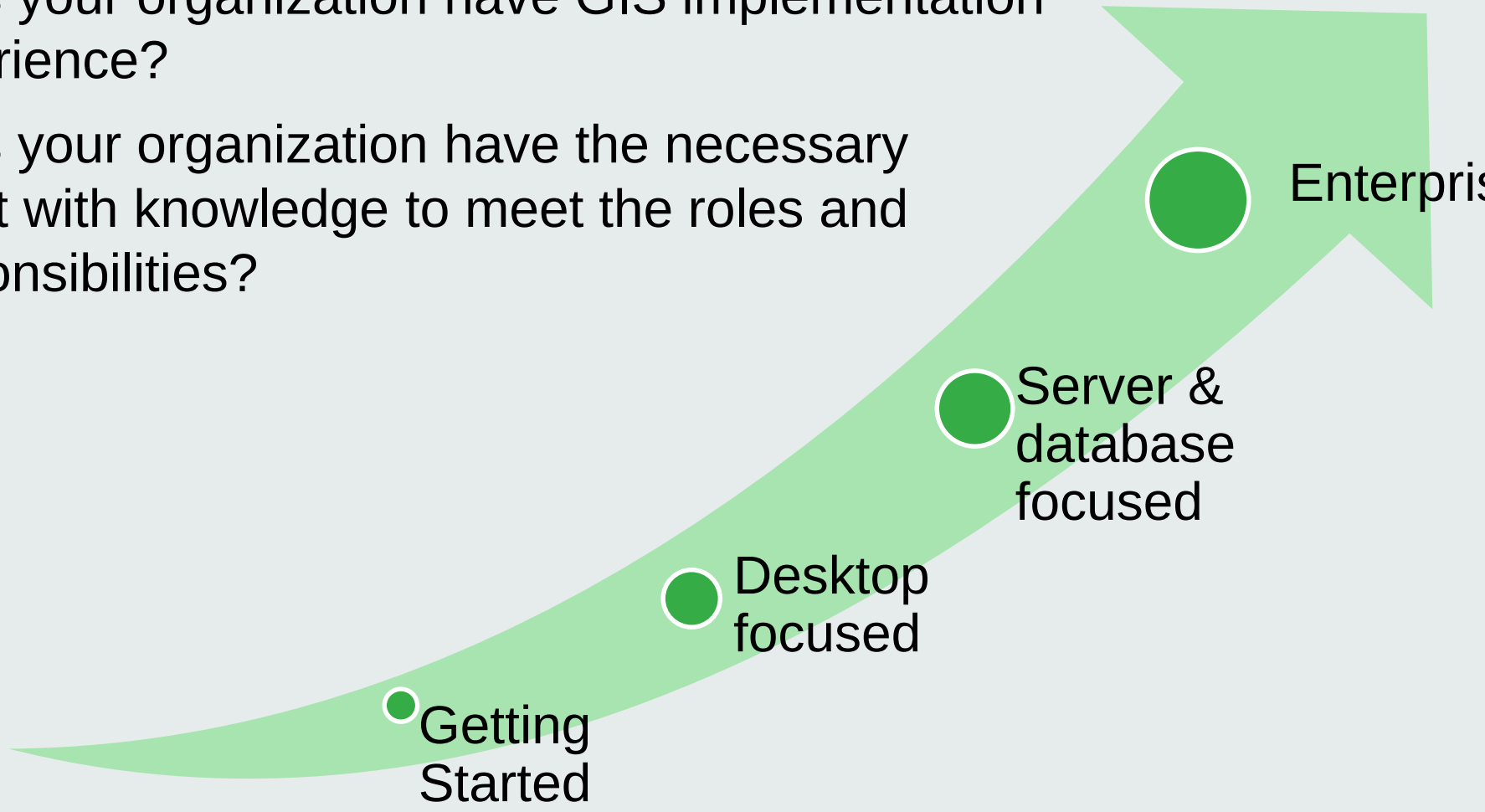


Capacity

Capacity of Human Resources, organization and staff
Historic usage and expected future growth
Monitoring and Testing
SLAs and the impact of over-capacity

Assess your experience

- Does your organization have enterprise software implementation experience?
- Does your organization have GIS implementation experience?
- Does your organization have the necessary talent with knowledge to meet the roles and responsibilities?



Assess your current IT/GIS capacity

- Hardware is cheap, experience is expensive
- Servers are easy, network is hard
- Beware of key components that you don’t control (NAS, firewalls, web accelerators…)
- Understand your vendor relationships
- Trust anecdotal information but verify

Assess your organization's willingness to change

- Who will be impacted by Enterprise GIS changes?
- How will they be impacted and how will they respond?
- What mitigation options are available for negative impacts?
- Plan your strategies carefully & engage
- An Agile approach can be a massive transition
- An Agile approach can yield earlier successes

Identify roles, resources and their availability

- What roles are needed to implement the Enterprise GIS?
- What would be their responsibilities?
- Identify in-house or contractor resources to potentially support the project as overseers or implementers
- Document roles, their responsibilities and candidates

GIS analyst

Account specialists

Database specialist

Technical lead

Subject matter experts

Program manager

Field collection specialist

Business Analyst

System architect

Project manager(s)

SOE programmer

Legal expert

Contract specialists

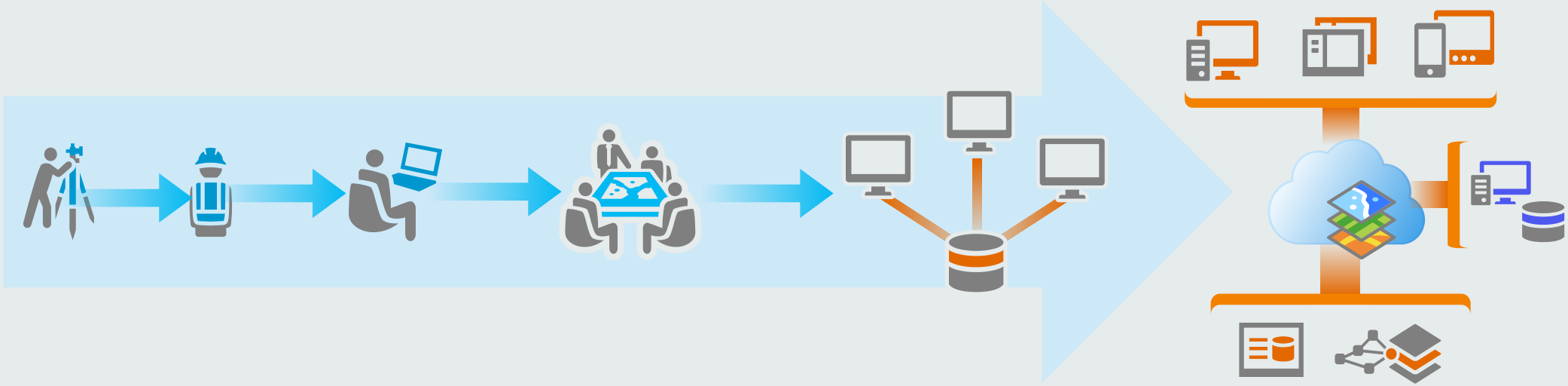
Formulate your capabilities and gaps

- List your assumptions
- List roles, responsibilities, potential resources and gaps
- List HW, SW and implementation gaps
- Identify mitigation options

Known Gaps Can Drive Requirements

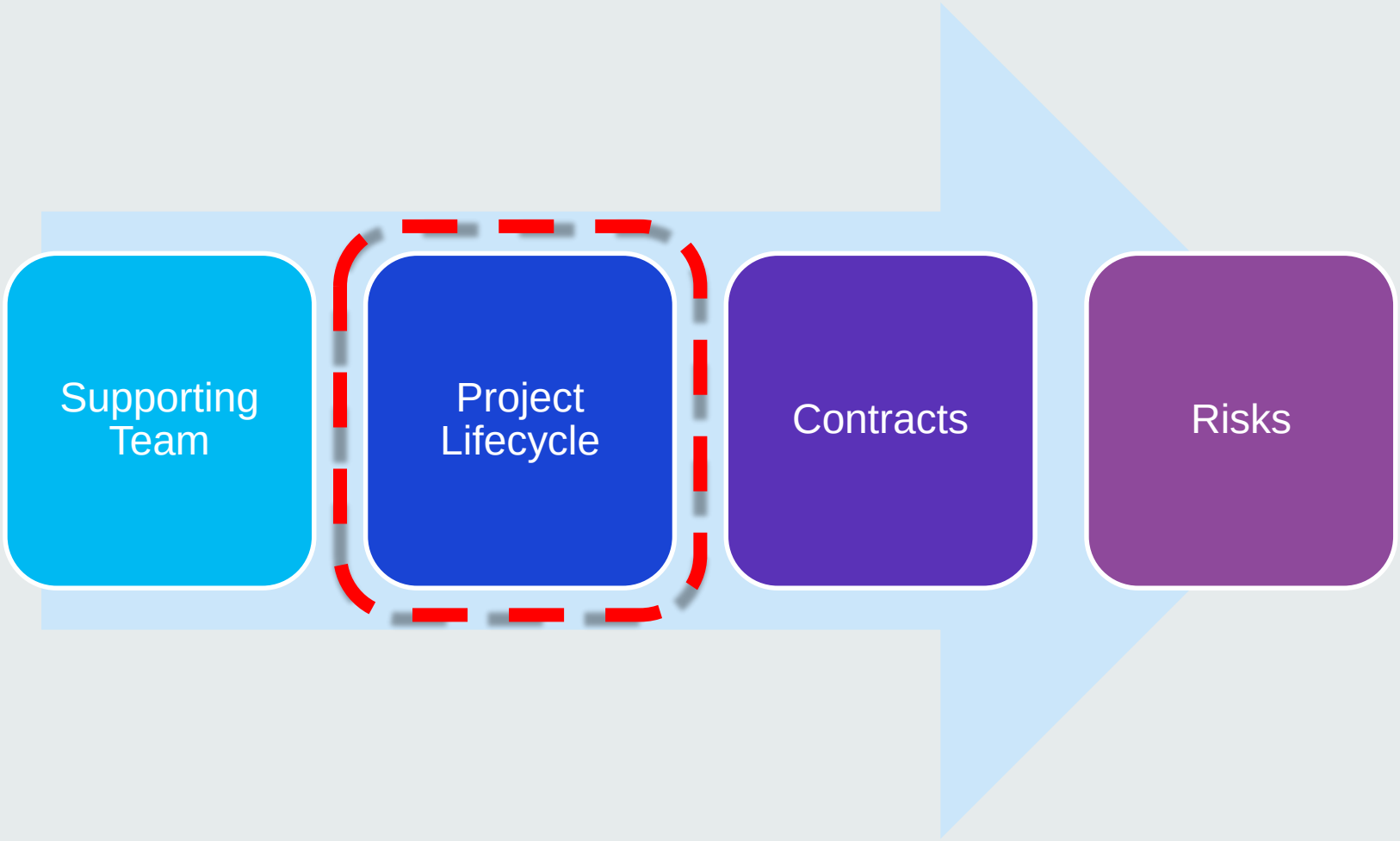
Test and Plan for Growth

- Test your GIS for performance (More this afternoon)
  - So you know when it's out of compliance
- Monitor system metrics and availability
  - So you know about problems before your customers do
- Create 1/3/5 year plans
  - To ease staff augmentation, hardware and software acquisition
- Stay current on software versions
  - Operating System, ArcGIS Platform, Database instances, 3rd party applications



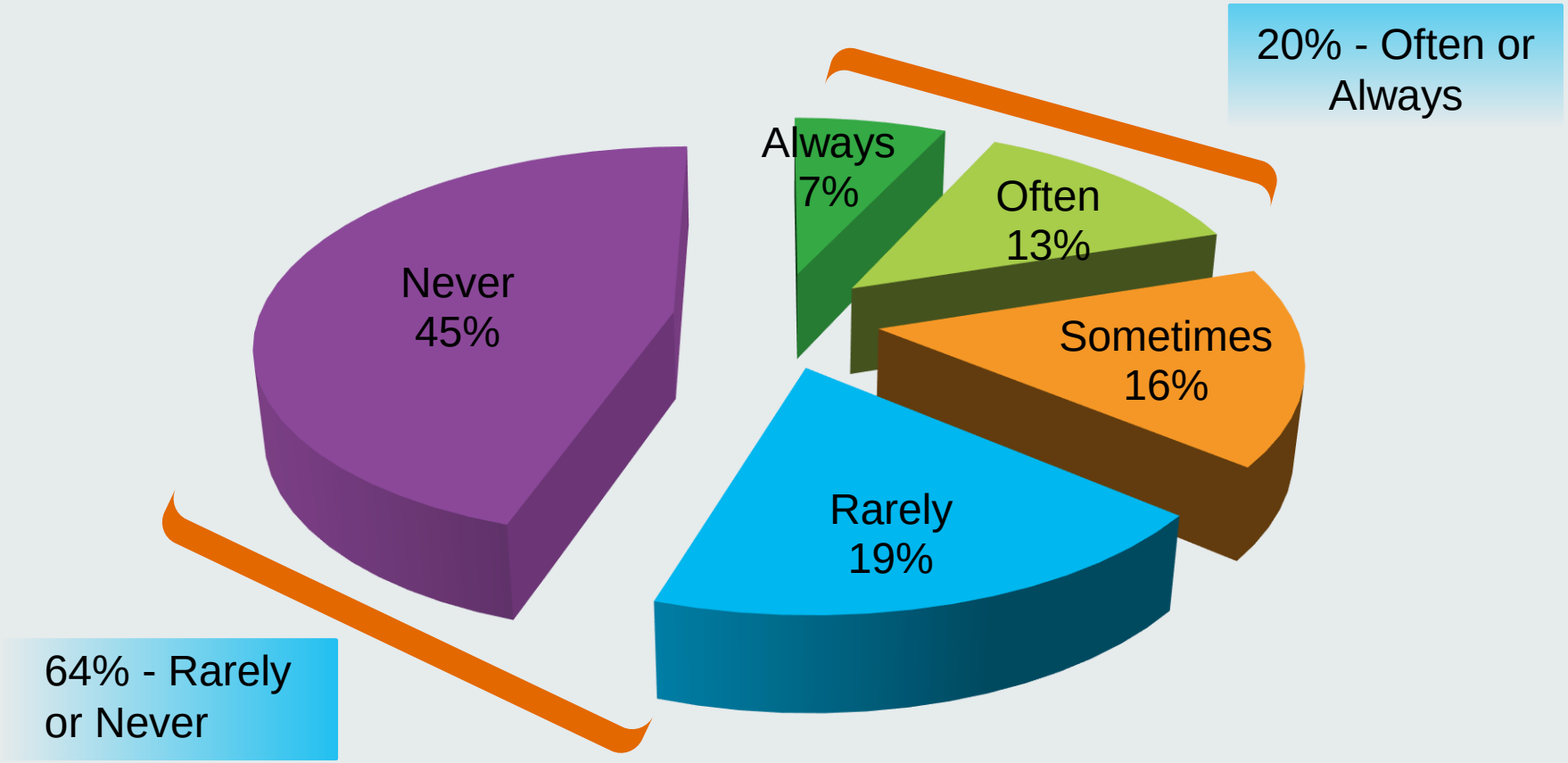
Known Gaps Can Drive Requirements

Assess and Plan



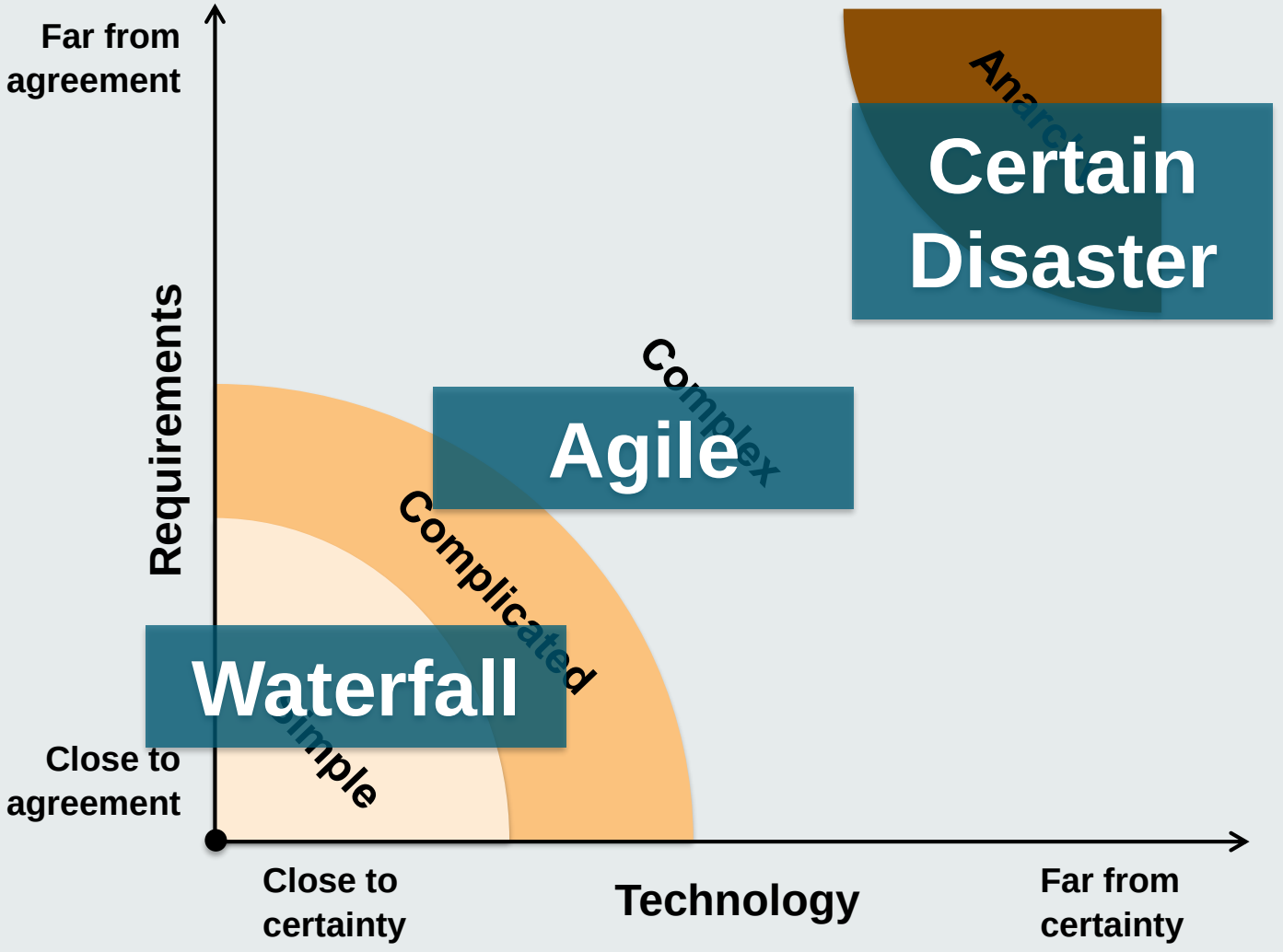
Build for Value

Requirements evolve over time

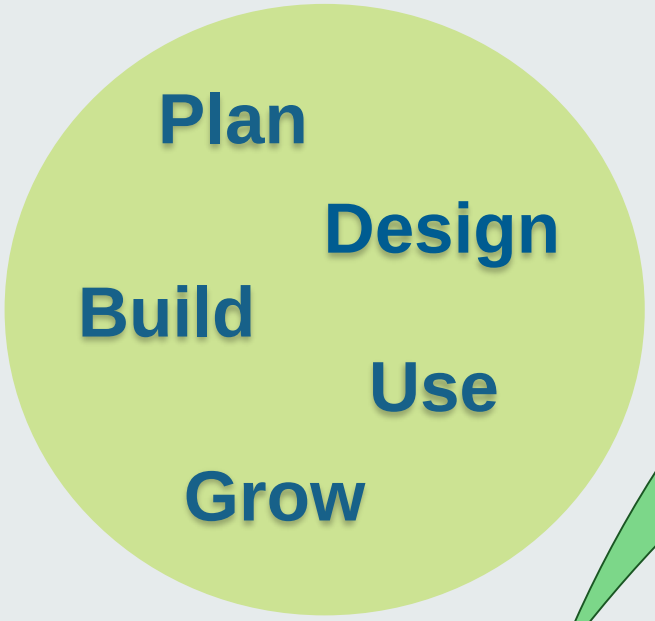


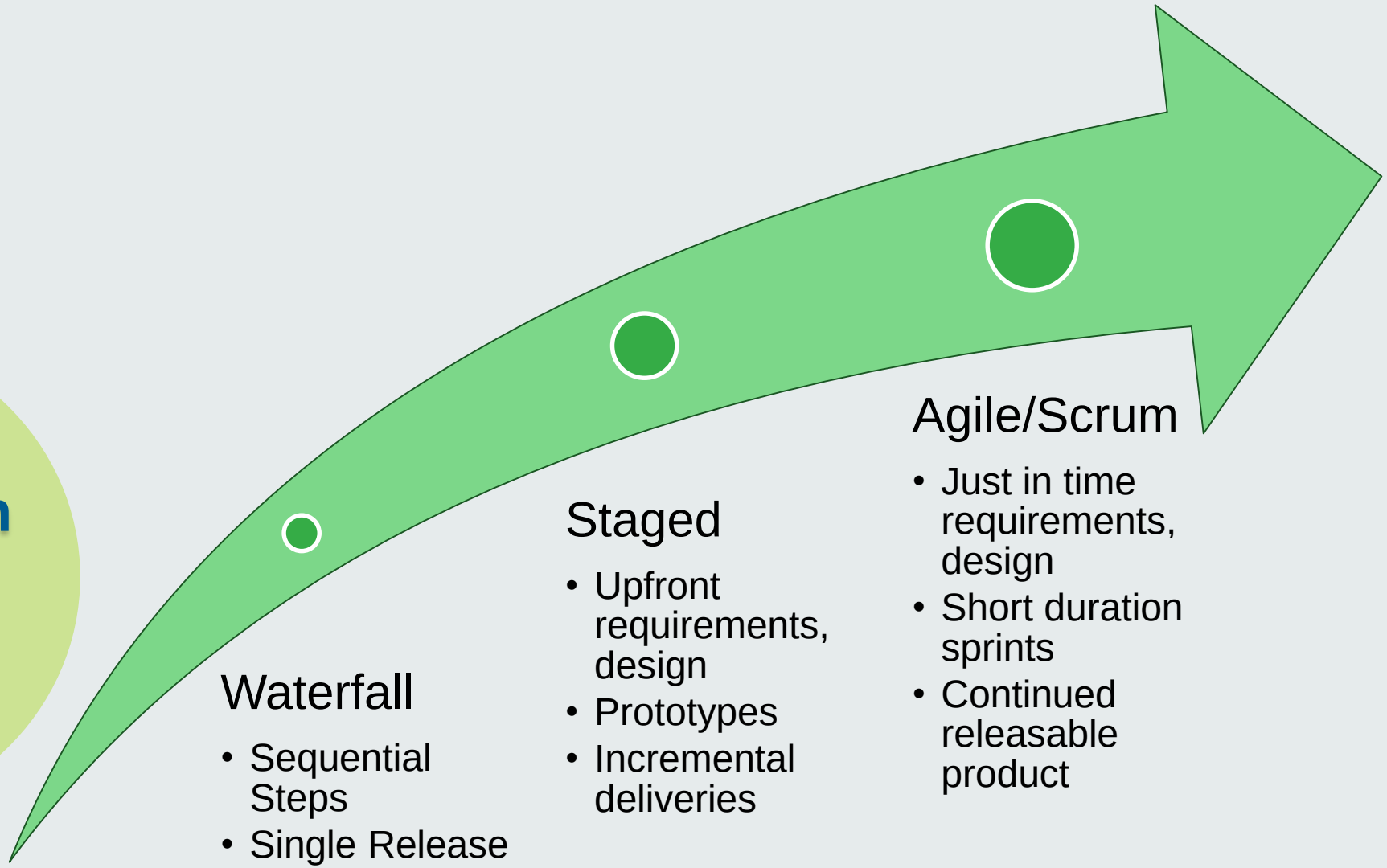
Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

How Requirements, Technology Affect your Choice of Project Lifecycle



Project Lifecycle Types





Waterfall

- Sequential Steps
- Single Release

Staged

- Upfront requirements, design
- Prototypes
- Incremental deliveries

Agile/Scrum

- Just in time requirements, design
- Short duration sprints
- Continued releasable product

Agile Manifesto

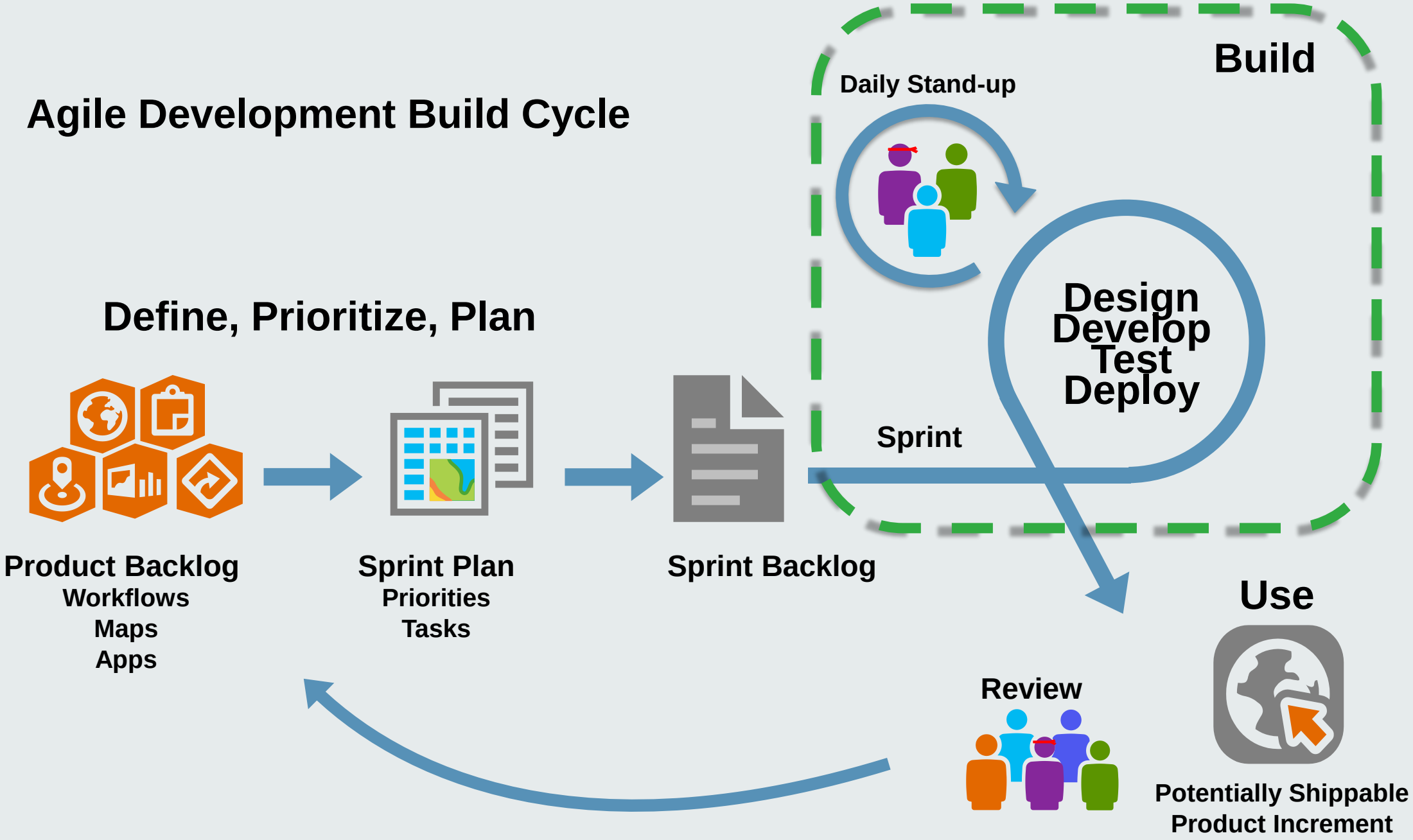
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

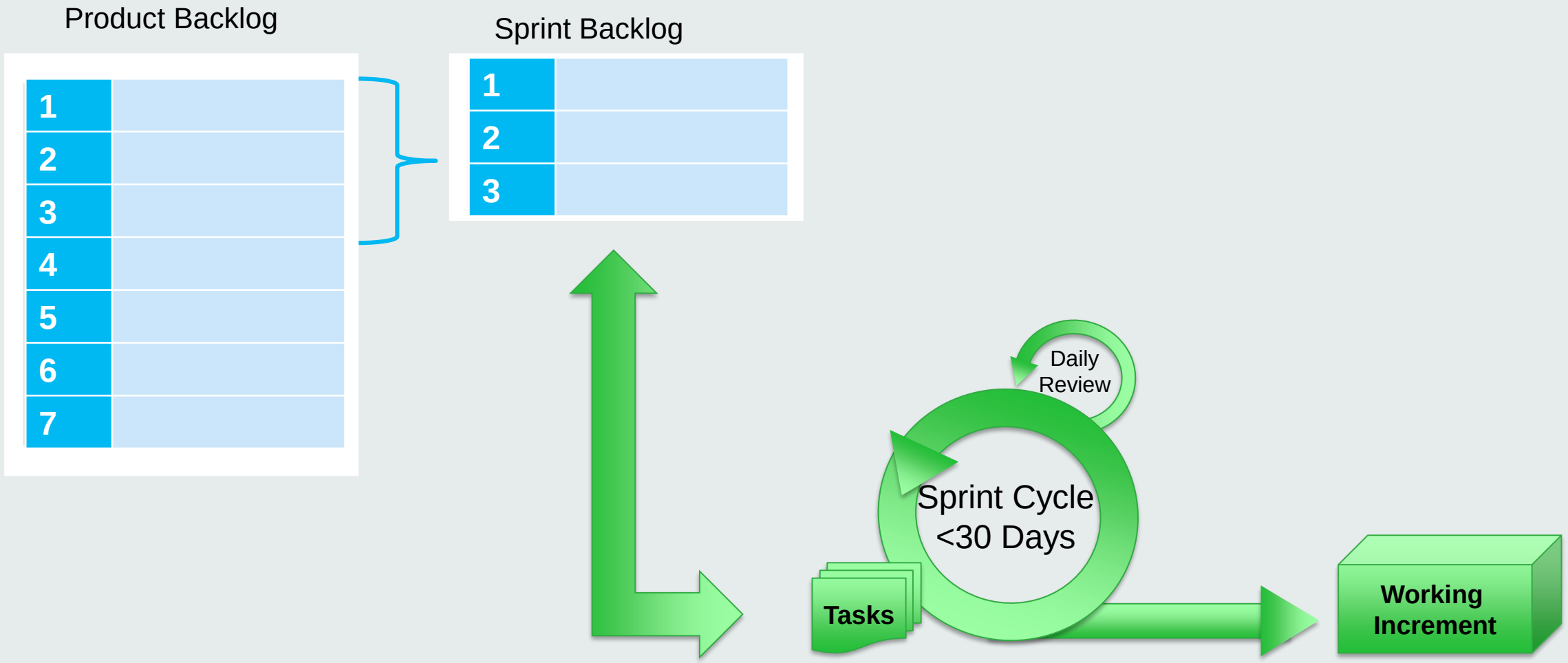
That is, while there is value in the items on the right,
we value the items on the **left** more.

Agile Manifesto - www.agilemanifesto.org

Agile Development Build Cycle



An Agile Sprint Lifecycle



When Do These Models Work Best?

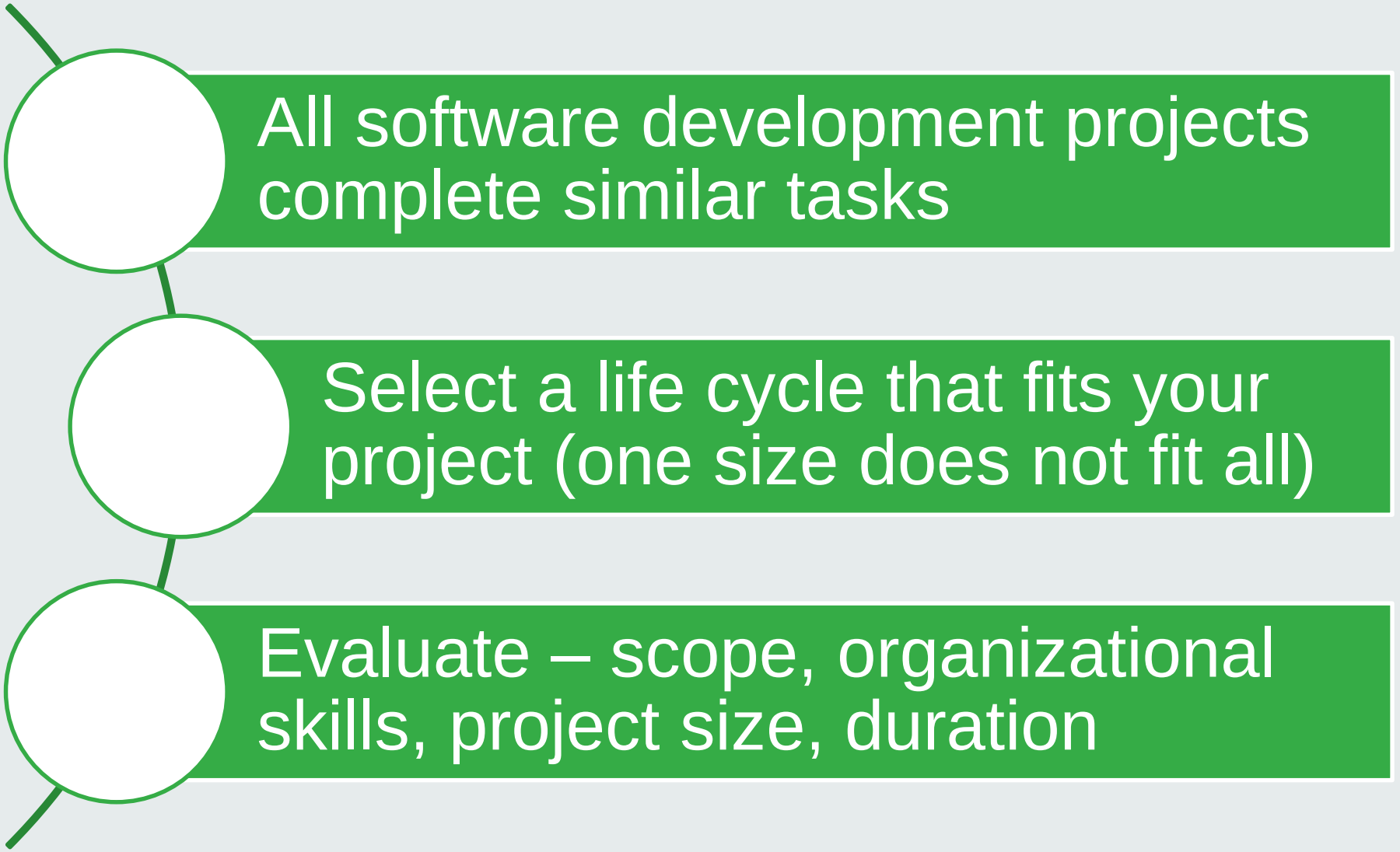
	Waterfall	Staged Delivery	Agile
Scope, Technology, Contract	• Clear requirements • Fixed deliverables • Single application	• Several applications • Prototypes expected	• Flexible scope, deliverables • One or several applications
Size, Duration	• Small size, short duration project	• Medium or large size, mid to long duration	• Any size or duration project
Capacity, Capabilities, Environment	• Limited capacity, resources, and environment • Frequent turnover on project team	• Capacity, resources, and environment to support multiple releases	• Customer EXPECTS collaboration • Stable, experienced project team

Exercise: Waterfall vs Agile

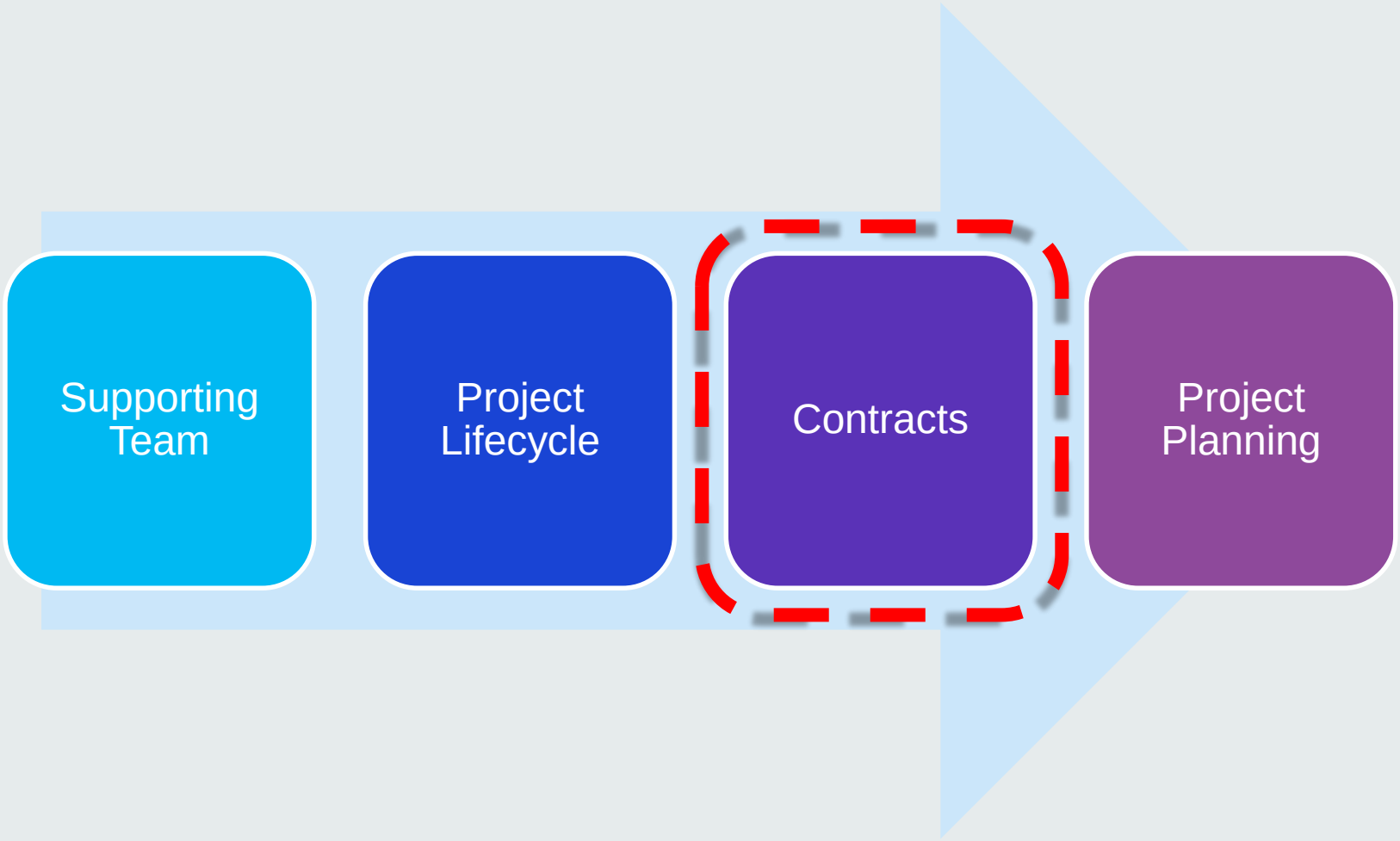
	Waterfall	Agile
Scope		
Planning		
Priorities		
Customer Participation		
Testing		
Development Team		
Project Management		

	Waterfall	Agile
Scope	Fixed and well -defined	Flexible, continuous refining
Planning	Extensive at the beginning allowing for faster development	Continuous and MUST be part of each iteration
Priorities	Set at the beginning	Constantly changing
Customer Participation	Can be minimal	Must be substantial
Testing	Not until the end (more bugs)	Constant (helps reduce bugs)
Development Team	Replacing resources is easier due to robust documentation	Must have strong skillsets and be adaptable
Project Management	Documentation and planning make managing “easier”	Strong PM skills required for scope management

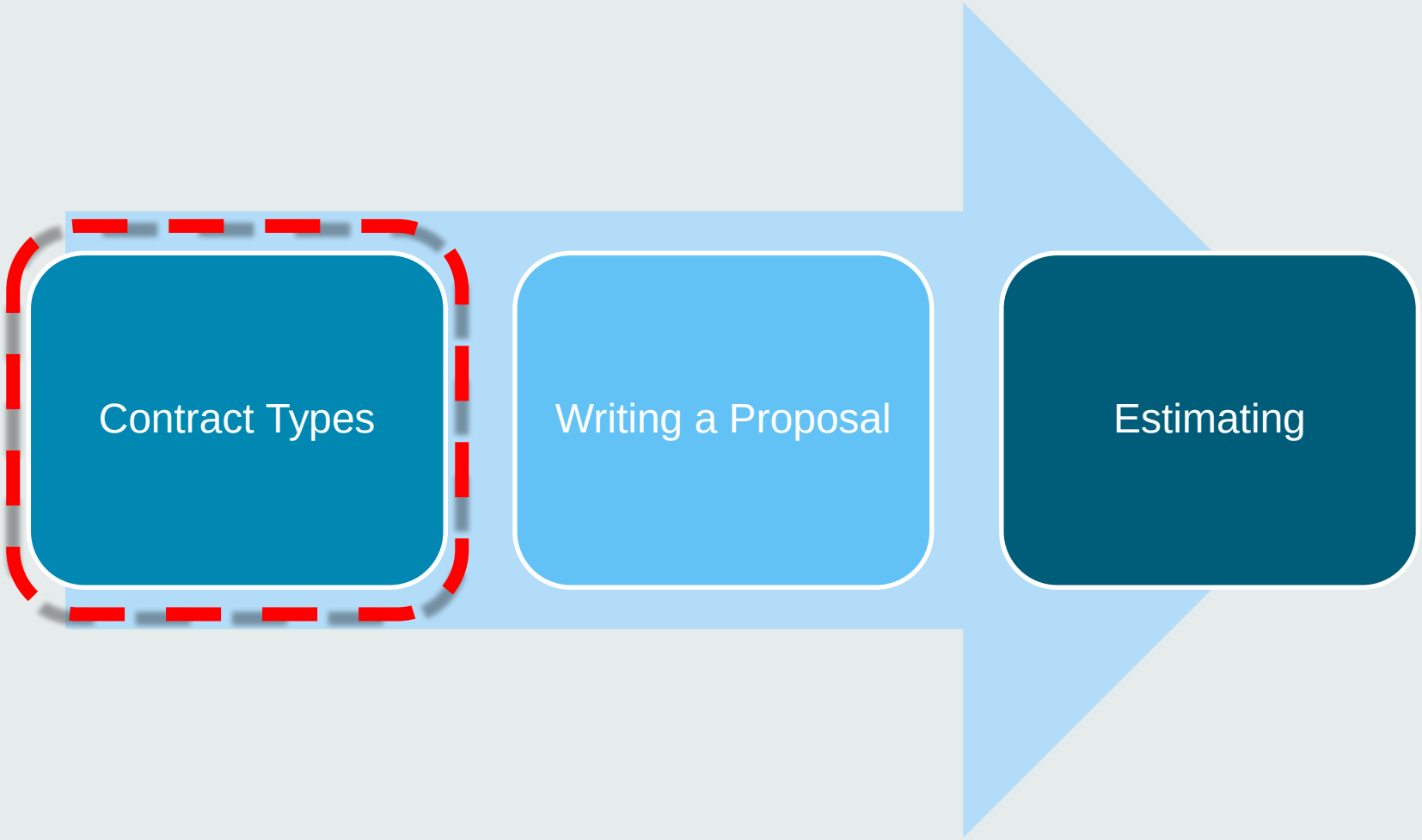
Key Takeaways



Assess and Plan



Assess and Plan



Contracts

- **Firm Fixed Price (FFP)**
  - **Contractors will bear the risk**
- **Time & Materials (T&M)**
  - **You bear all the risk**

Contract questions	FFP	T&M
Do you know what you want?	A lot	Little
How Much is Managing Risk Worth to You?	A lot	Little
How Important is Flexibility to You?	Little	A lot

Contract Support

- **Are you going to engage Contractors?**
- **If so what would be the Contract type**
  - **Firm Fixed Price (FFP); or**
  - **Time and Material (T&M)**

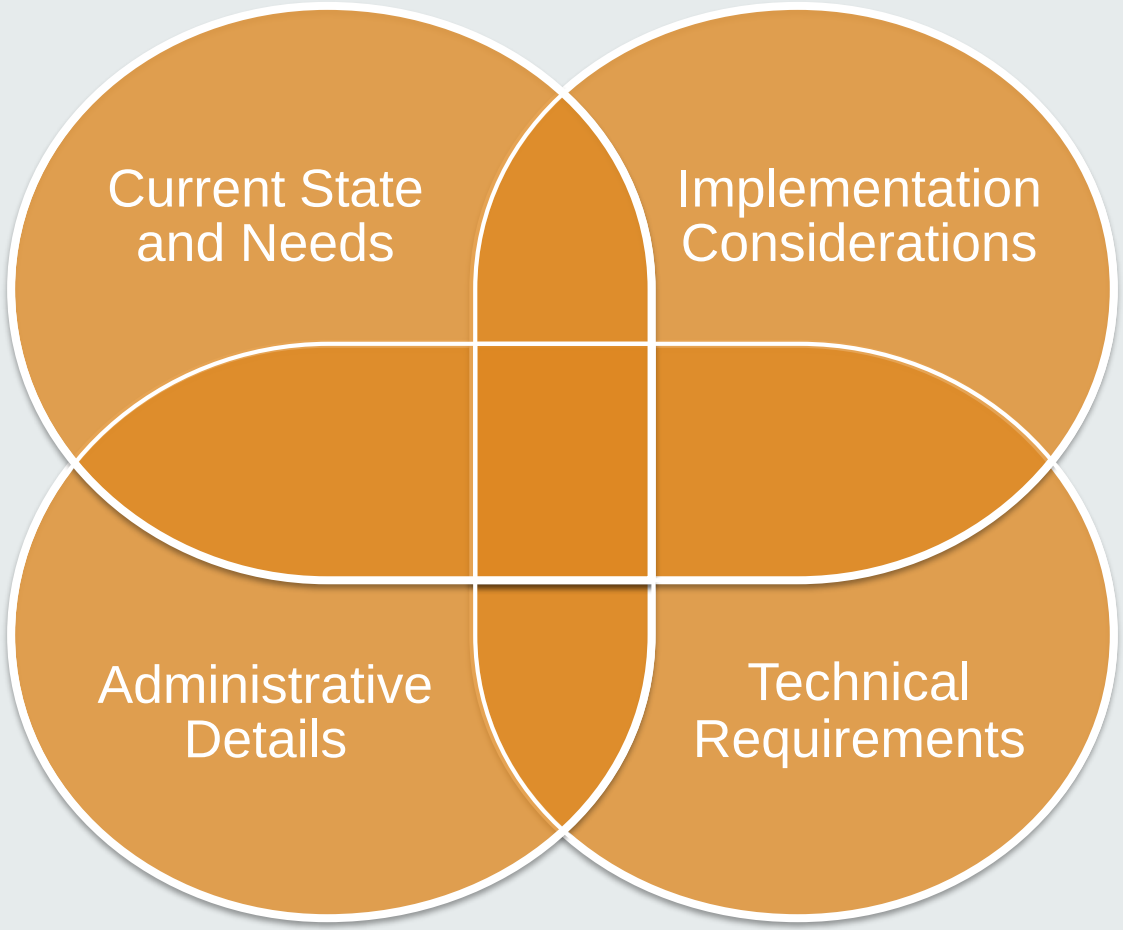
You?

Contractors?

Writing an RFP for Subcontractors

What should you include in an RFP?

Not Just Technical Details!

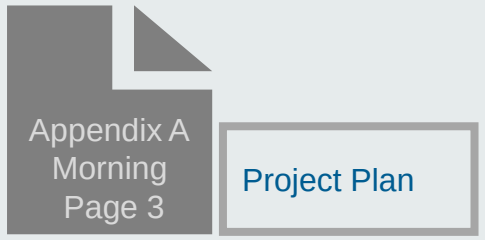


Good RFPs Strike a Balance

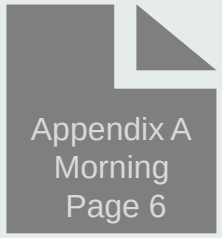
RFP Process (from the Vendor perspective!)

So What Should Be In Your Enterprise Plan?

- Vision statement
- Key workflows
- Approach – life cycle, budget and staffing
- Key things you need from the rest of the organization
- Schedule – overview of key milestone target dates
- Assumptions, dependencies and risks
- Appendices – WBS, roles, key points of contact, etc.

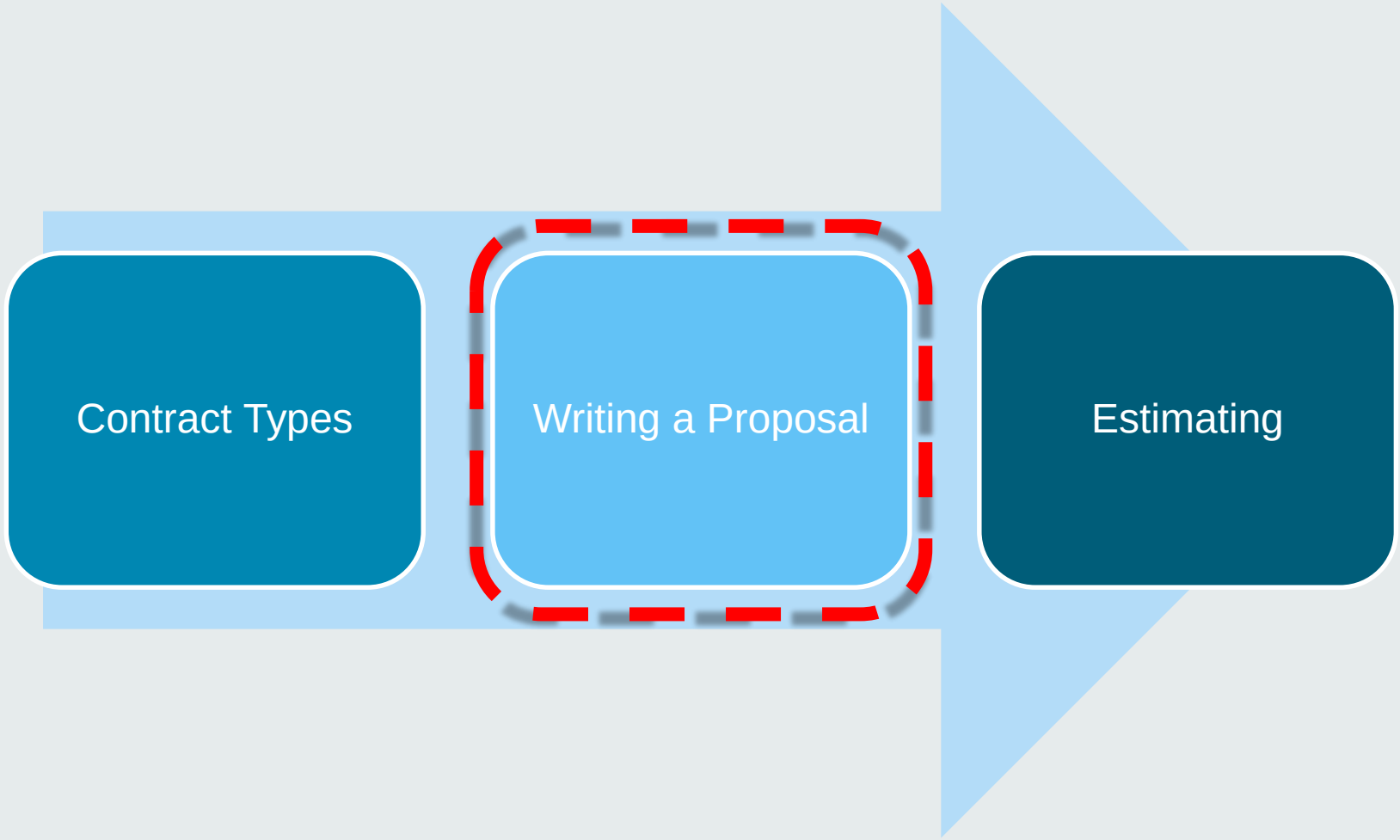


Project Plan



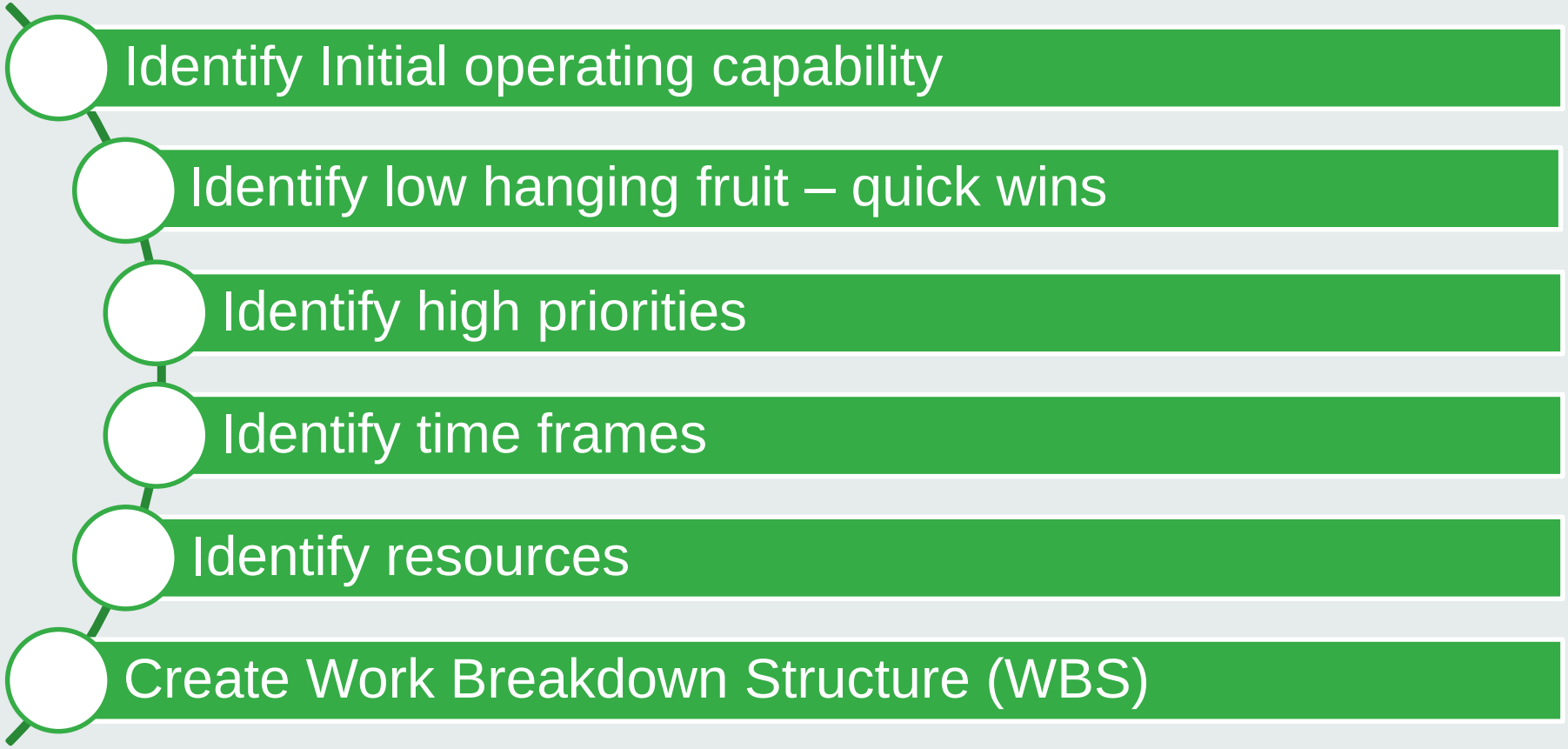
Project Risk Matrix

Assess and Plan



Proposing is Part of Planning

How to turn information into an initial plan



Proposal Considerations

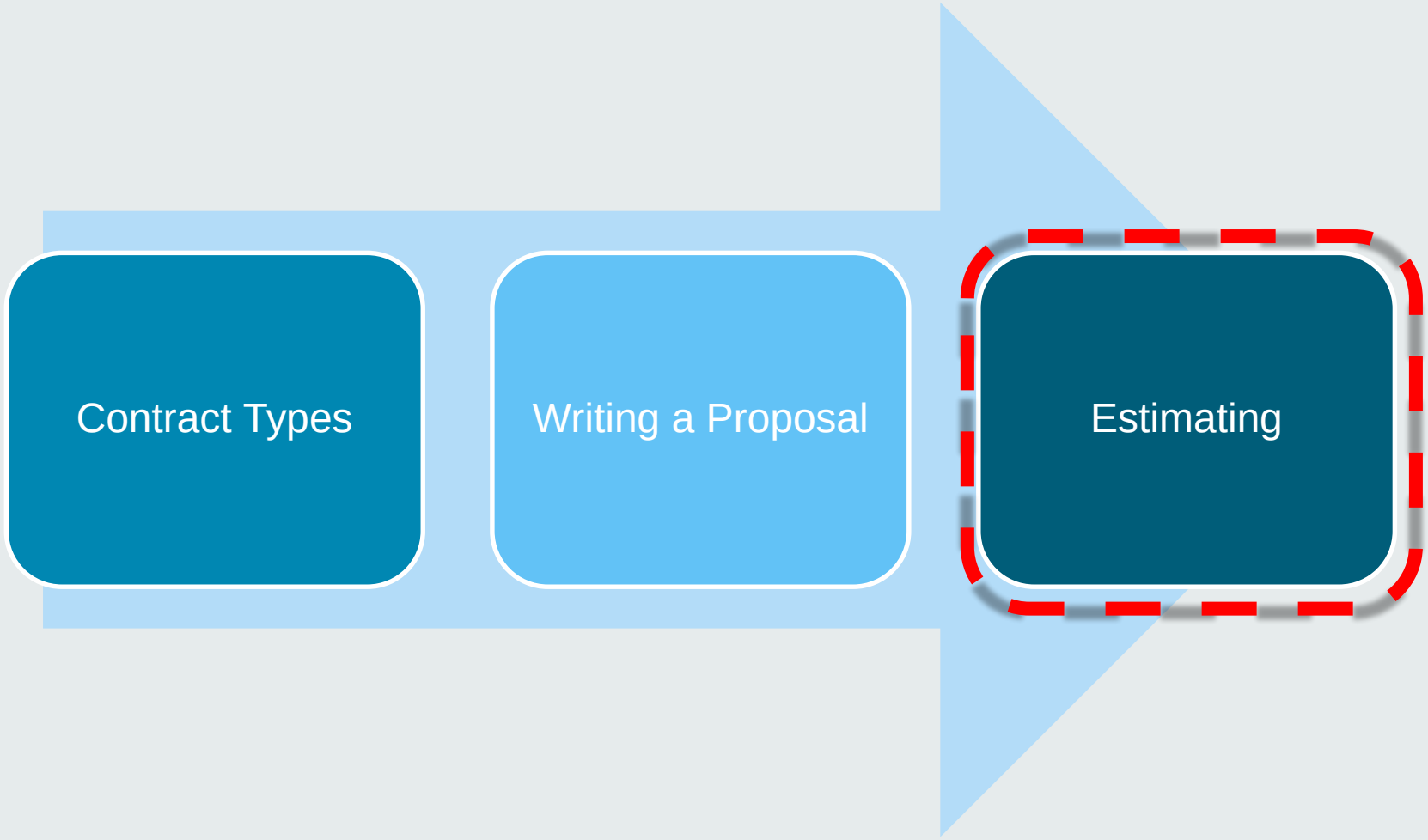
Source: Point of Beginning: "How to Submit a Successful GIS RFP Response"; Valerie King, October 1, 2016; http://www.pobonline.com/articles/100594-how-to-submit-a-successful-gis-rfp-response

Writing a Proposal

Focus on the Key Areas…

- Executive Summary
- Solution Overview
- Scope of Work
- Schedule

Assess and Plan



Big Picture

Epics

System Architecture

Database Design

Widget 1

Story point is a arbitrary measure used by Scrum teams. This is used to measure the effort required to implement a **story**. In simple terms its a number that tells the team how hard the **story** is. Hard could be related to complexity, Unknowns and effort. In most cases a **story point** range is1,2,3,5,8,13,21,34,45

Story Points

21

34

34

45

21

Story Points

21	34	34	45	21

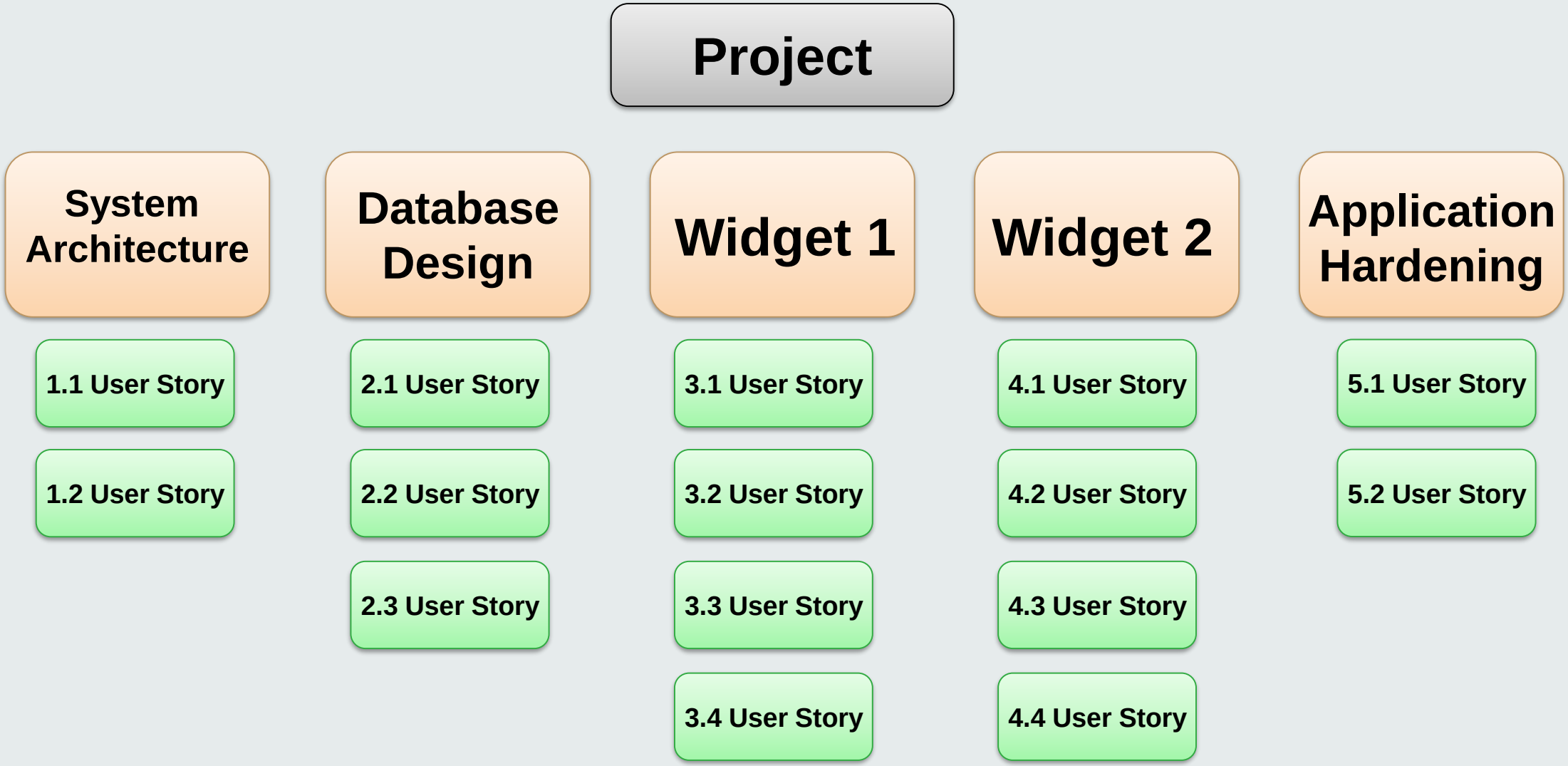
x 8

Hours

168	272	272	360	168

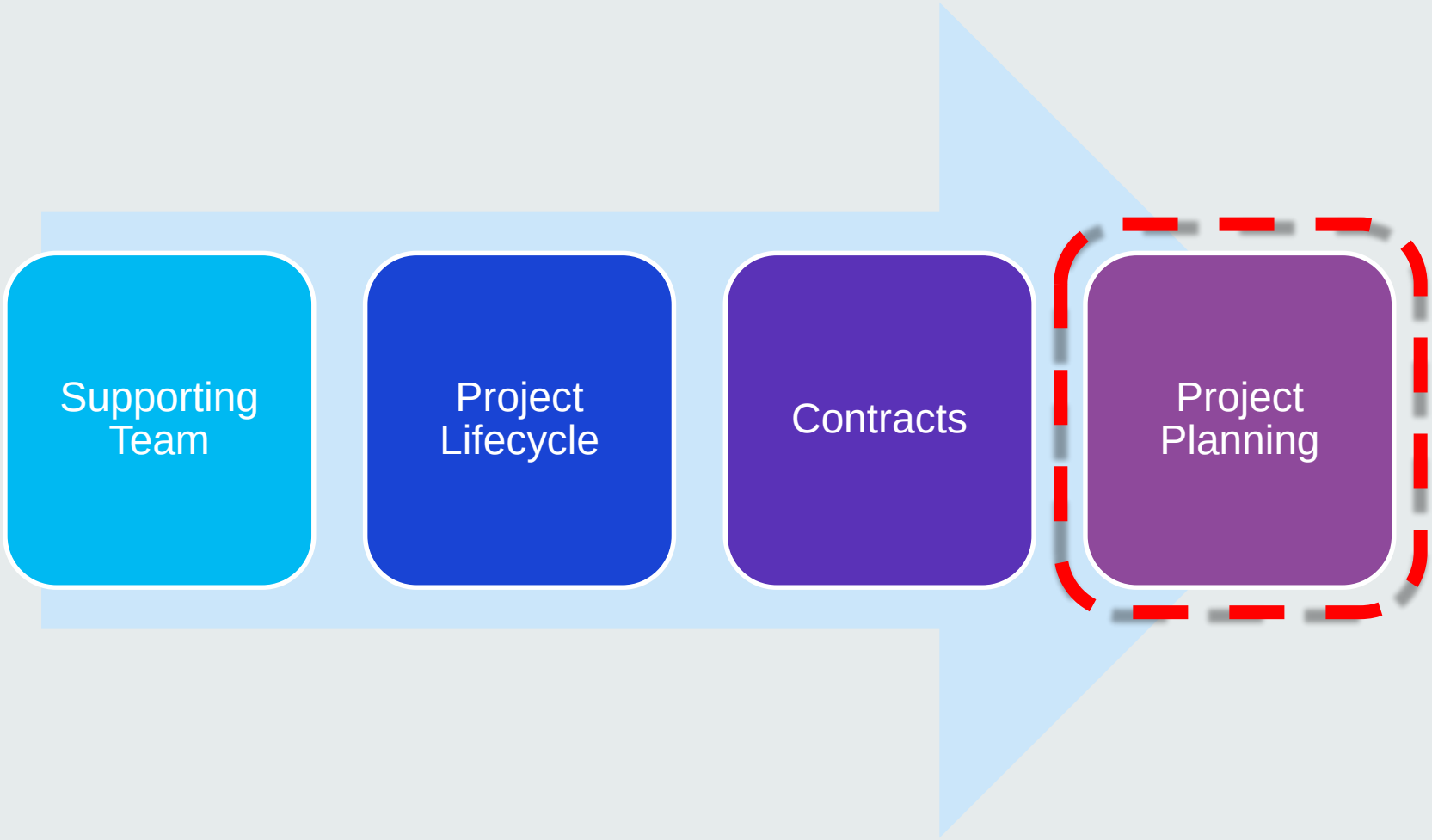
Estimating Sheet

Activity	Scrum Master	Product Owner	Developer	Analyst	System Admin	Total
System Architecture	16	16	0	16	120	**168**
Geodatabase Design	24	24	0	184	40	**272**
Widget 1	24	24	176	48	0	**272**
Widget 2	20	20	240	80	0	**360**
Application Hardening	40	16	84	24	4	**168**



Work Breakdown Structure

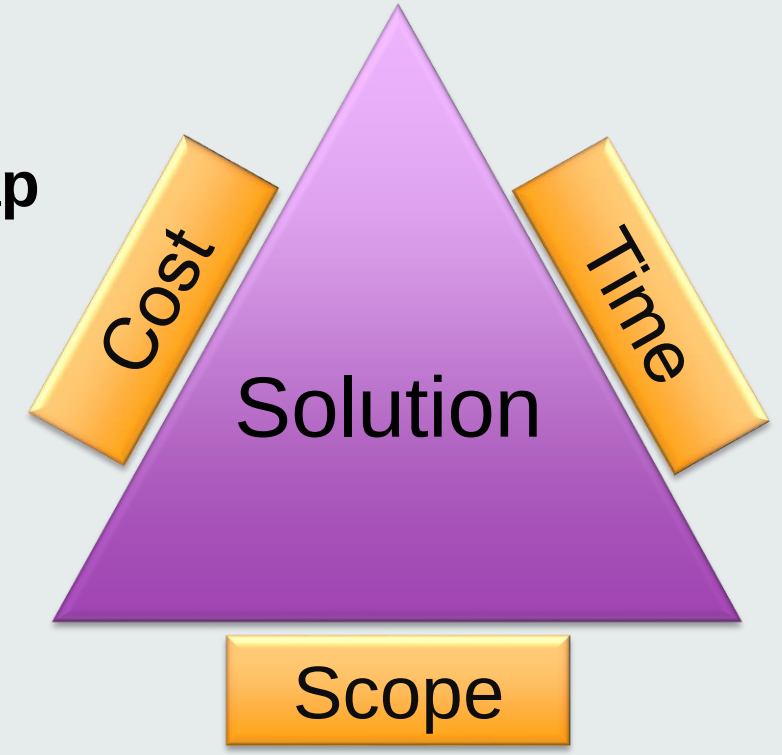
Assess and Plan



Writing a Project Plan

Why is a Project Plan Critical?

- **Defines the Project Execution Roadmap**
  - **What you are building**
  - **Key Timing, sequence of events**
  - **Resources**
  - **Strategy and Communications**
- **Defines when you are done**
  - **Quality expectations**
  - **Acceptance Criteria**



Document Priorities

- **Group and list priorities:**
  - **Key workflows**
  - **Key data**
  - **Key IT needs**
  - **Staffing needs**
  - **Schedule**
- **Rank importance**

List Requirements (High Level)

- **High level functional requirements**
  - **Business processes**
  - **Output**
  - **Communication**
- **Non-Functional**
  - **Technology**
  - **Capacity**
  - **Maintenance and support**
  - **Policies and procedures**

Implementation Team

Project size and complexity	Project complexity increases with size Large projects will require more dedicated roles
Conflict of interest, separation of duties	If you are the Developer, should you be testing? If you are the Project Manager, should you be the Release Manager?
Skills and expertise required for the task	Do you have domain knowledge to do requirements? Do you have the qualifications needed for testing or release management?
Best use of time	Should a senior manager test or focus on other things?

Setup WBS

- Work Breakdown Structure (WBS) is aligned with the project plan to support project tracking and monitoring
- WBS is decomposing the project into phases, deliverables and work packages
  - Manageable work elements
- Derived from the Statement of Work (SOW)
- All branches in the WBS should be well defined otherwise they represents a risk in scope

Appendix A
Morning
Page 8

Sample WBS Elements

Accuracy of charging

Too few

Ease of reporting

Ease of charging

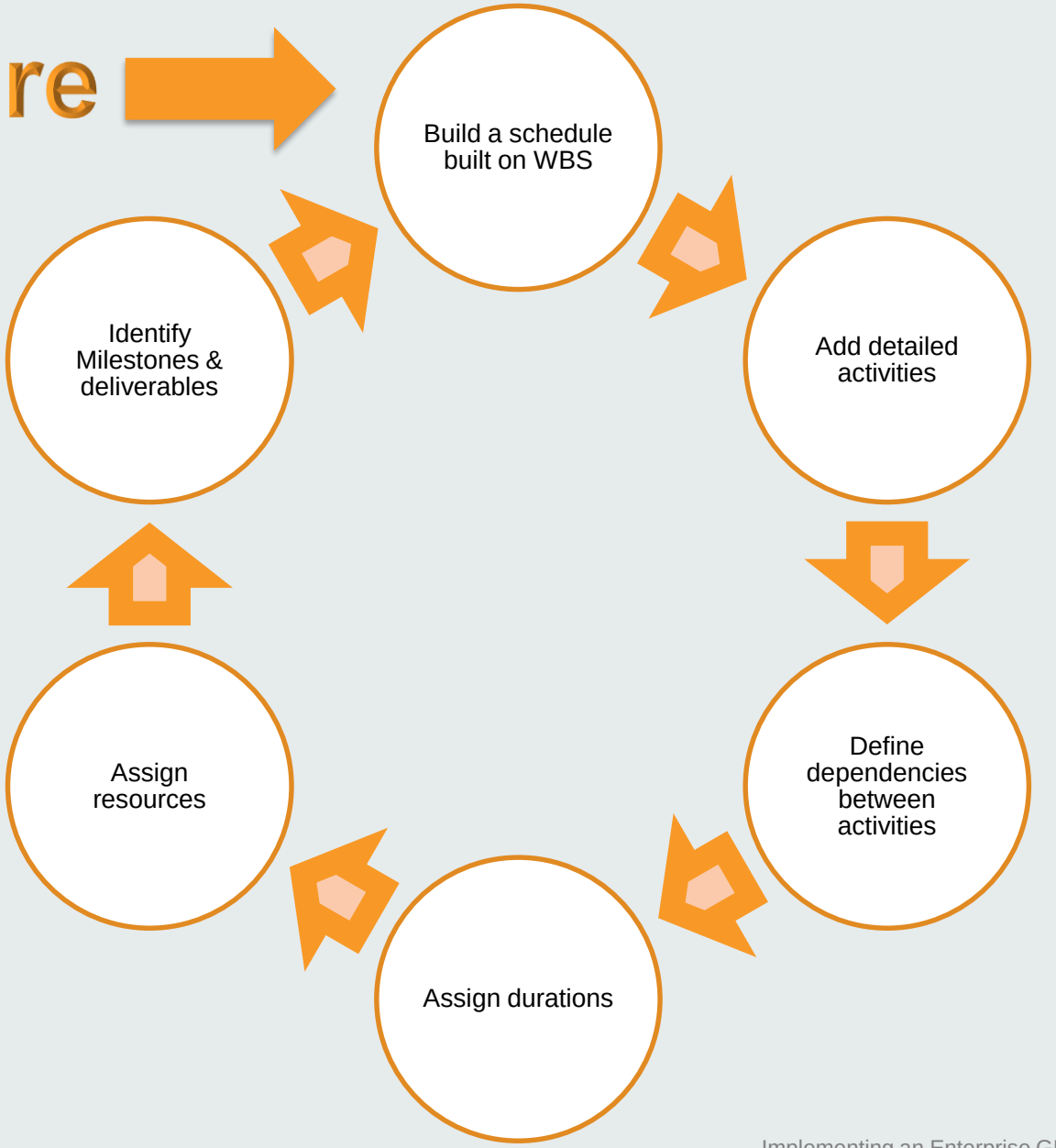
Too many

Complexity

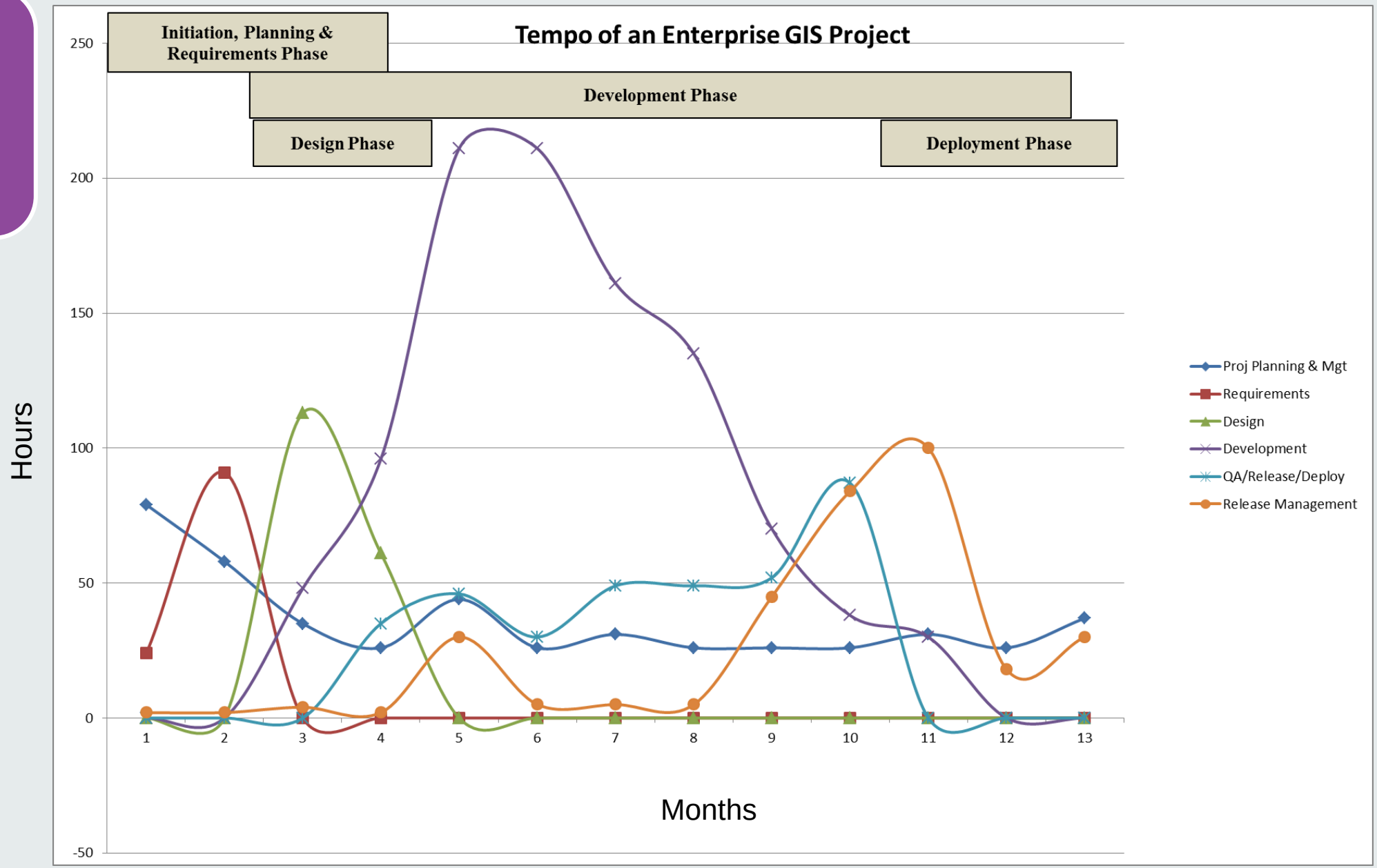
A Good WBS...

Develop Baseline Schedule

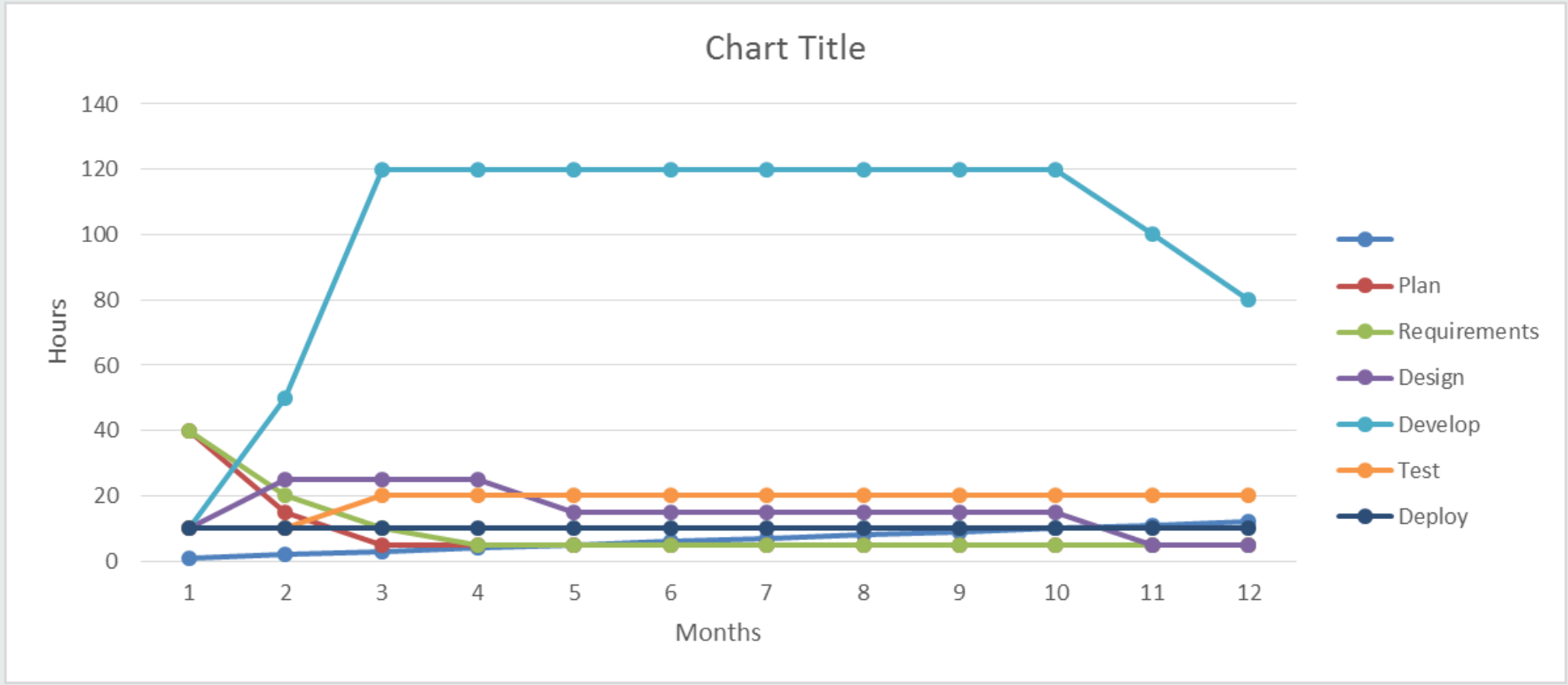
Start Here



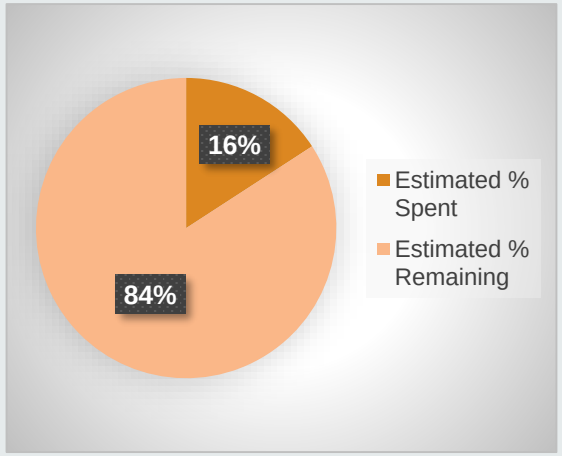
Waterfall Approach

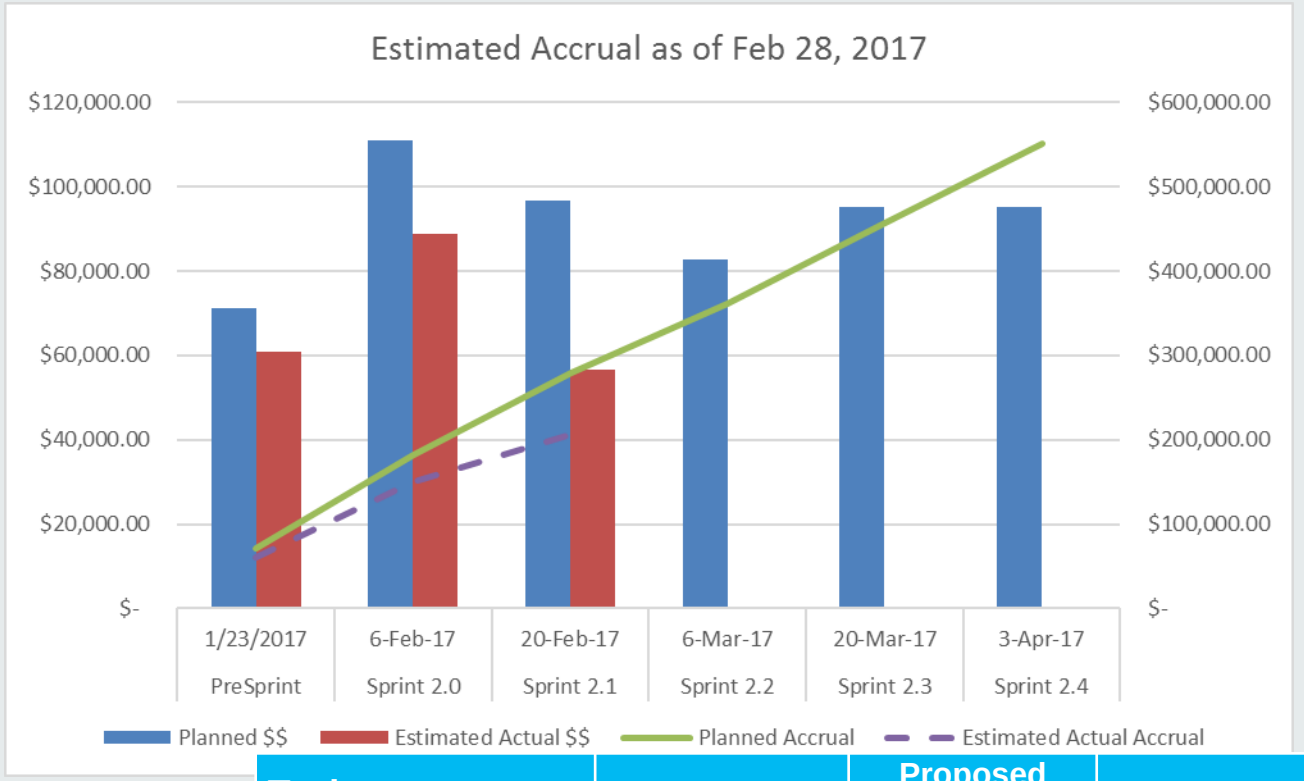


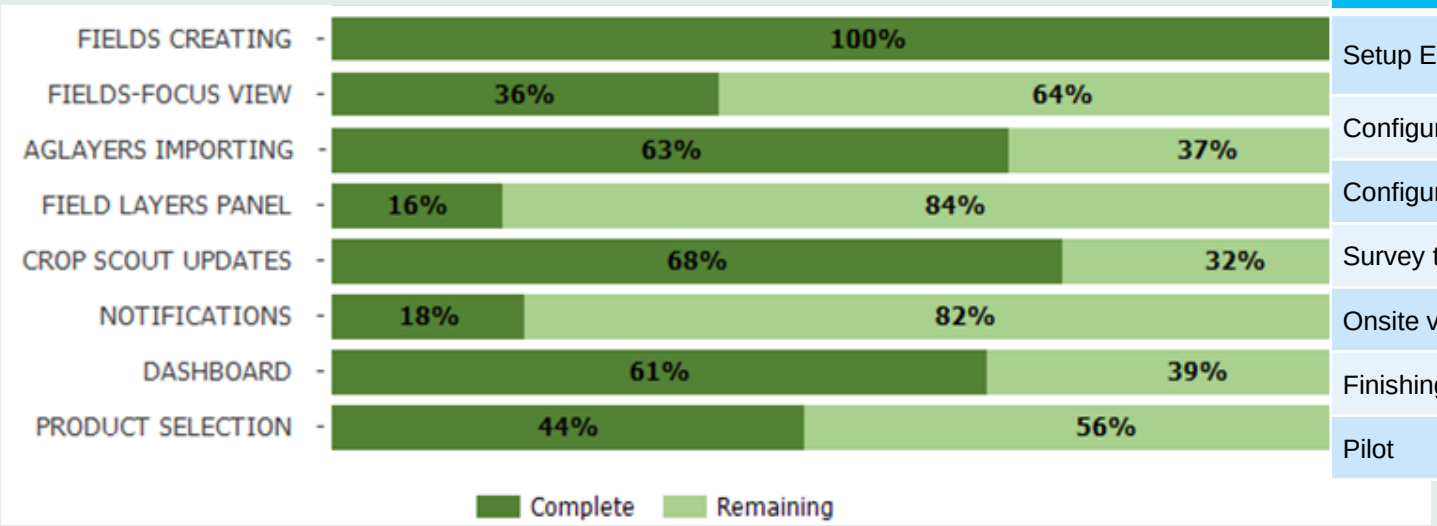
Agile Approach



Reporting







Task	Proposed Start	Proposed Finish	Status
Setup EMCS environment	4/10/17	4/14/17	
Configure Portal	4/14/17	4/28/17	
Configure Surveys	4/14/17	4/28/17	
Survey tie-in	4/24/17	5/19/17	
Onsite visit	5/9/17	5/11/17	
Finishing touches	5/15/17	5/19/17	
Pilot	After 5/22		

The Importance of Managing Risk

https://www.youtube.com/watch?v=Wac3aGn5twc

- **Watch the video**
- **Think about:**
  - **What went wrong?**
  - **What would you do differently?**
- **There will be a test!**

Video Review

- **What went wrong?**
  - **Execution**
    - **No business rhythm**
    - **Poor communication**
    - **Over allocated resources**
  - **Control**
    - **Changing scope**
    - **No schedule**
    - **Undefined budget**
    - **No risk management**

Let’s look at this more closely…

Risk Management

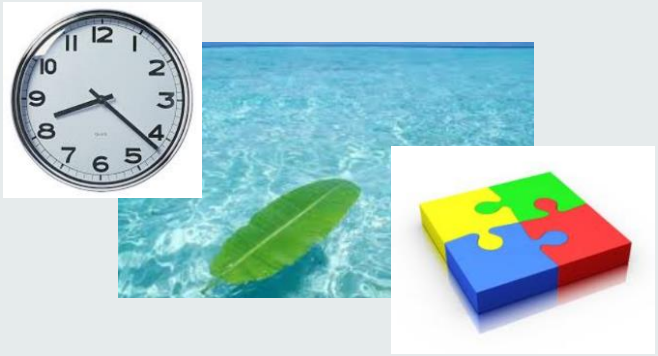
What happens when there is no risk management?

Purpose of Risk Management

What is the purpose of risk management?

Identify

RISK



Reduce

Sources of Risk Exercise

What are sources of risk?

Examples:

- **Impossible schedule**
- **Loss of funding**
- **Natural disasters- e.g. Hurricane Katrina 2005**

Take a piece of paper

Write down one idea of your own

Take five minutes

Sources of Risk

What are sources of risk?

No project management

Too many projects

Not enough resources

Too many changes

Change in government or regulation

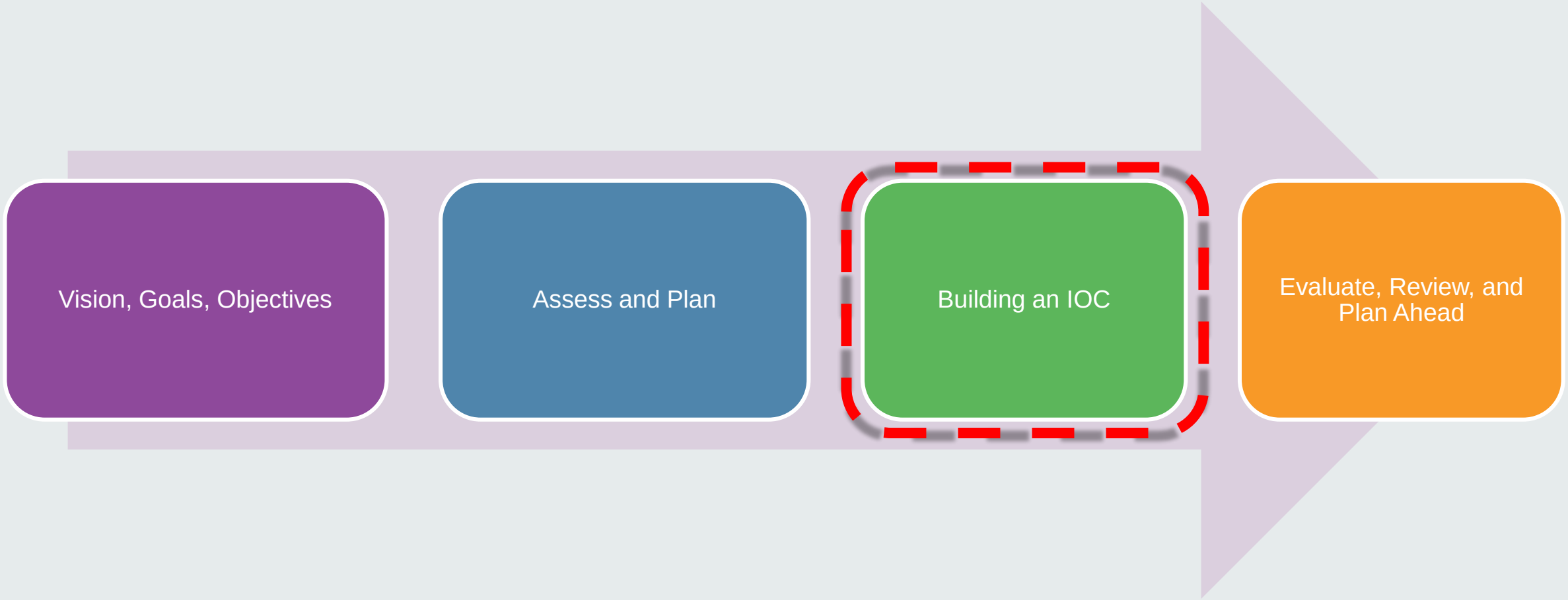
Legal issue

Currency rate fluctuations

Technology changes

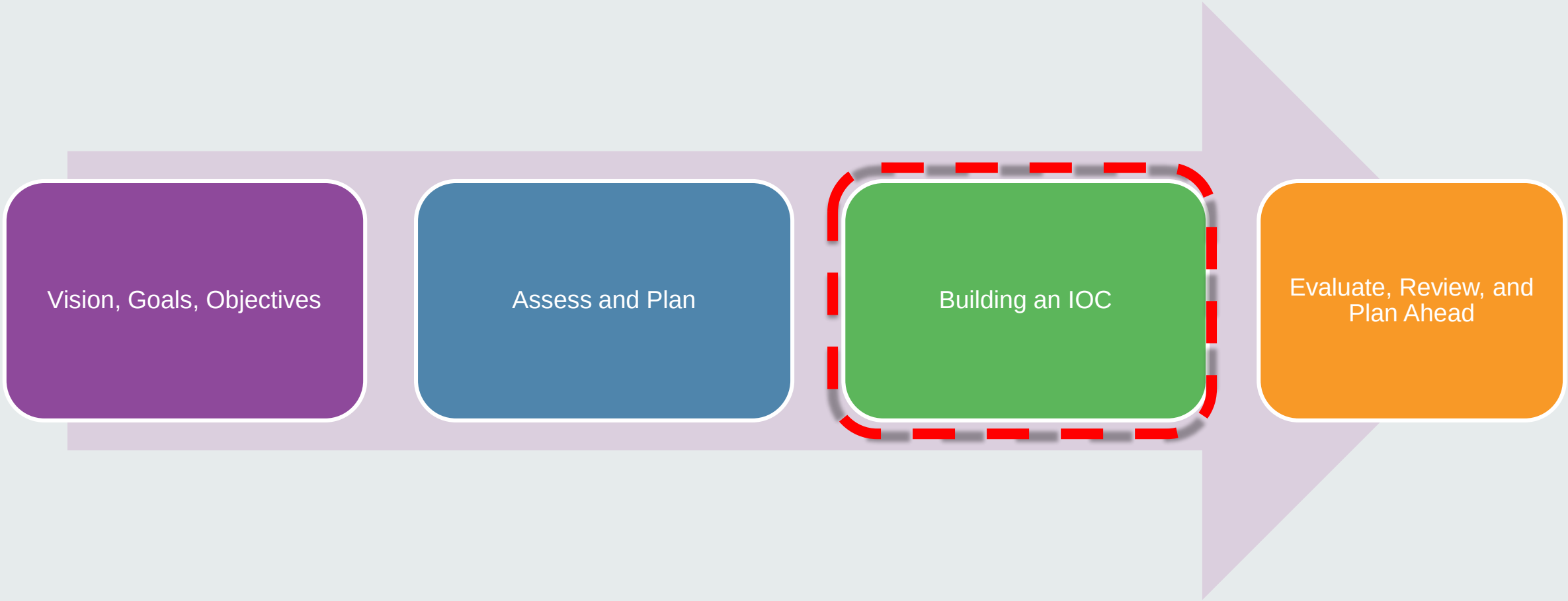
And many more…

Assess the Following Components

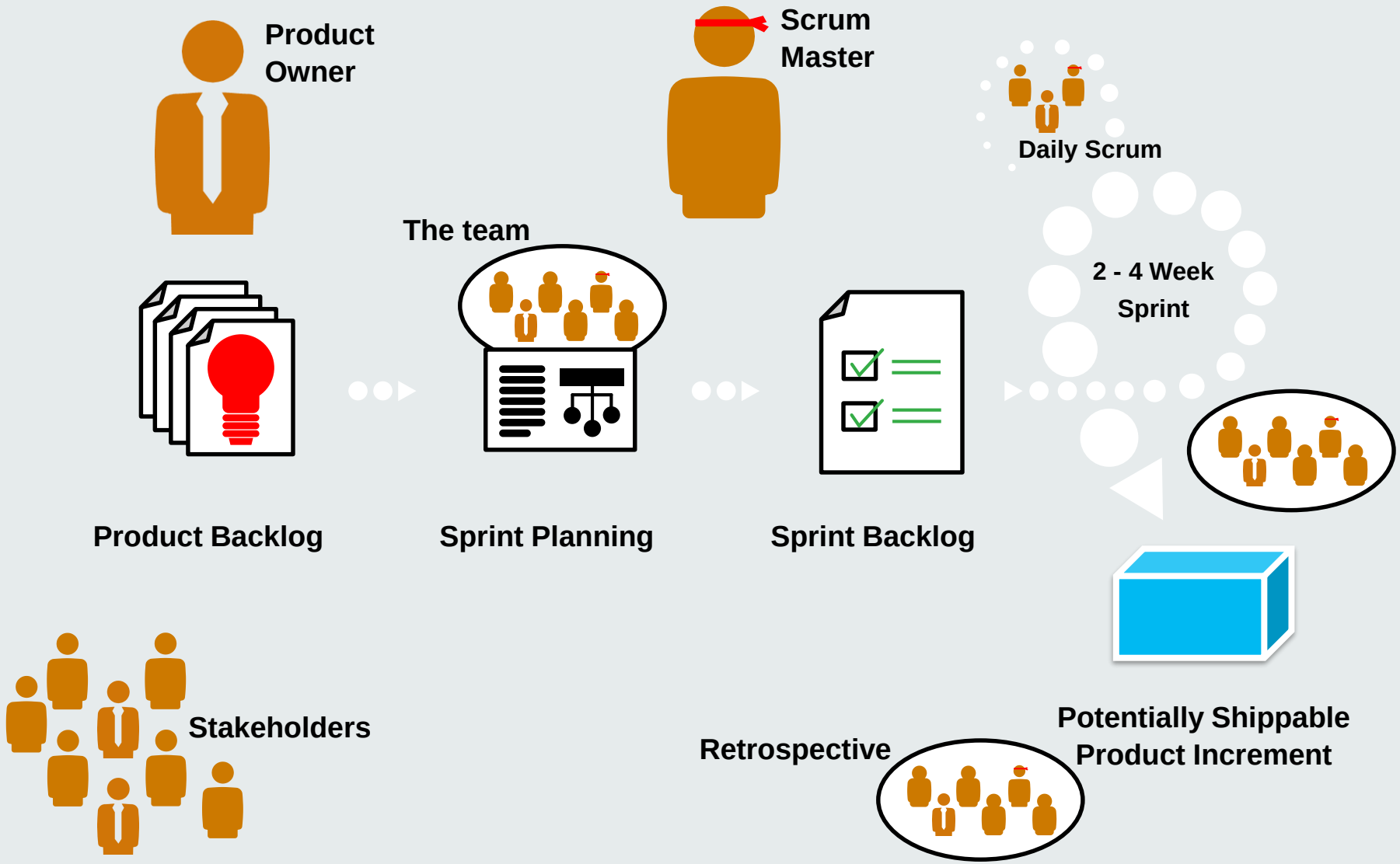


Lunch Break
11:30am - 1:00pm

Assess the Following Components

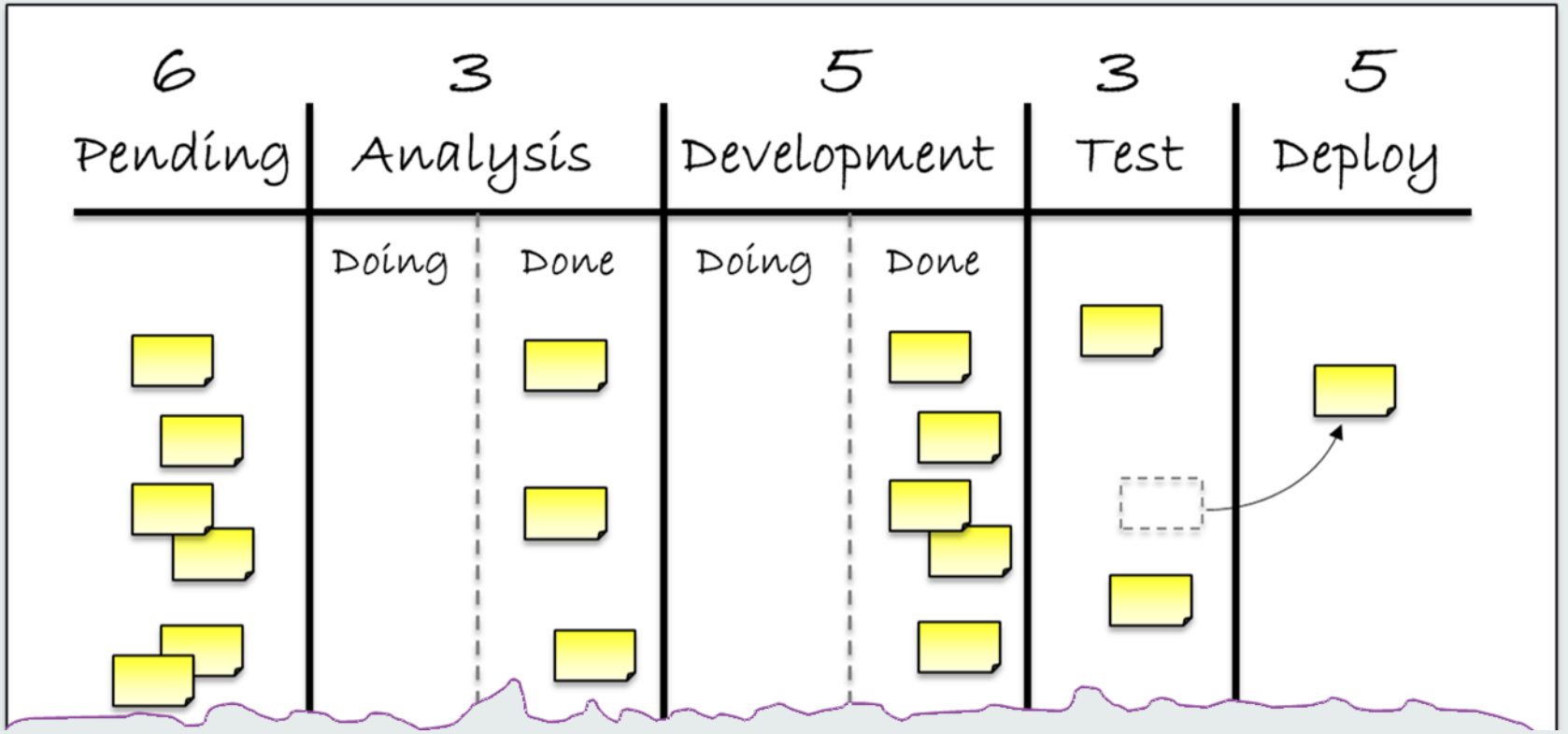


Scrum Sprint Cycle



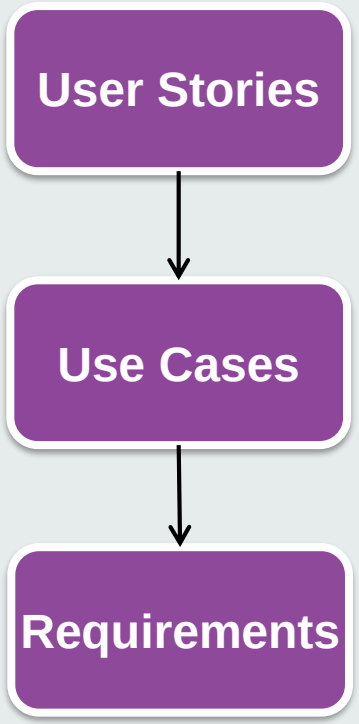
KanBan Approach (Still Agile, just not Scrum)

- No defined iterations
- No defined roles
- Direct communication with customer
- Limit your work-in-progress
- Visualize your work
- Ever-changing backlog with on-the-fly prioritization



User Stories, Use Cases, Requirements…

How do we relate them?



Narrative texts that describe an interaction of the user and the system, focusing on the value

Conceptual level

Criteria to which the system or business must adhere

Functional requirement

Series of interactions by the user with the system and the response of the system

Technical requirement

Appendix A
Afternoon
Pages 10-19

User Story and Requirements Examples and Documentation

Types of Requirements

Business, Functional/Technical, COTS

Business Level

- Streamline processes
- Reduce costs
- Increase productivity
- Improve communication between user groups

Functional Level

- Enhance capabilities to accomplish work tasks
- Store metadata to support historical metrics
- Use location information to harness the power of GIS

Performance/Quality Level

- Must be faster than the old system
- Must be on software that is still supported
- Must be built with future growth allowable

Identifying opportunities of improvement

A good user story uses the “INVEST” model:

Independent. *Reduced dependencies = easier to plan*

Negotiable. *Details added via collaboration*

Valuable. *Provides value to the customer*

Estimate-able. *Too big or too vague = not estimate-able*

Small. *Can be done in less* than a week

Testable. *Good acceptance criteria*

As a [role], I can [feature] so that [benefit]

As a field representative,
I want to collect information offline
so that data can be collected
in remote locations.

Use stories to facilitate a **conversation** with the team and with the users…

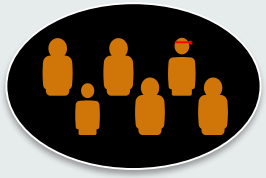
Product Owner

Scrum Master

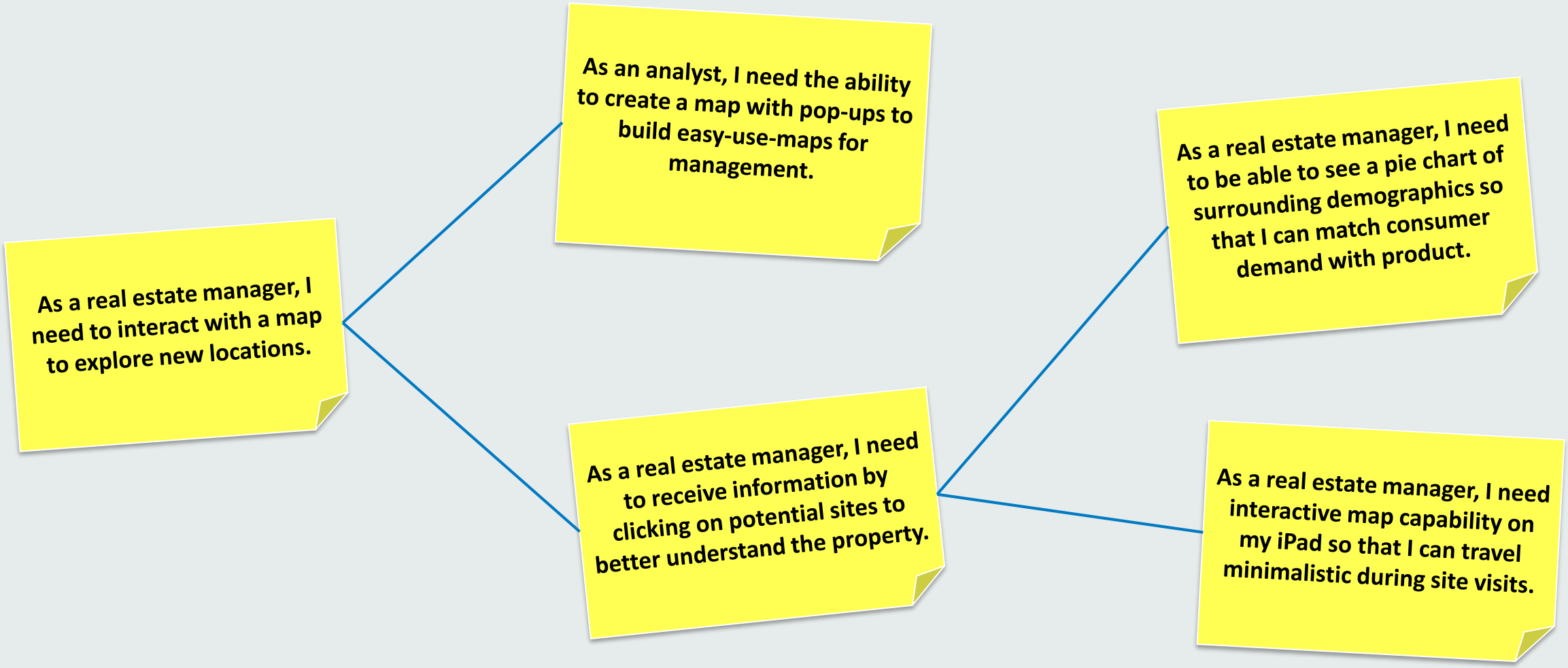
Hey! Something to look at, talk about, comment on, react to…

Stakeholders

The team



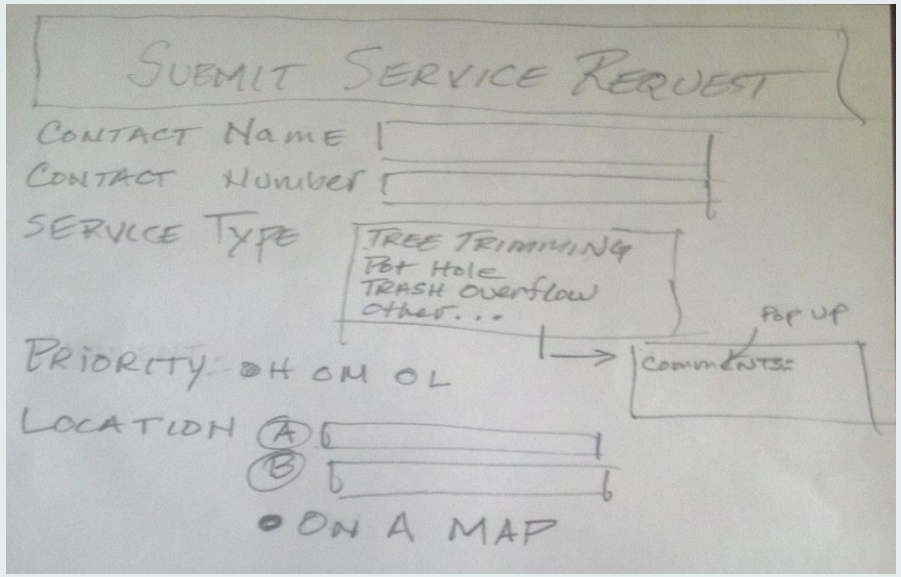
Progressively decompose your stories…

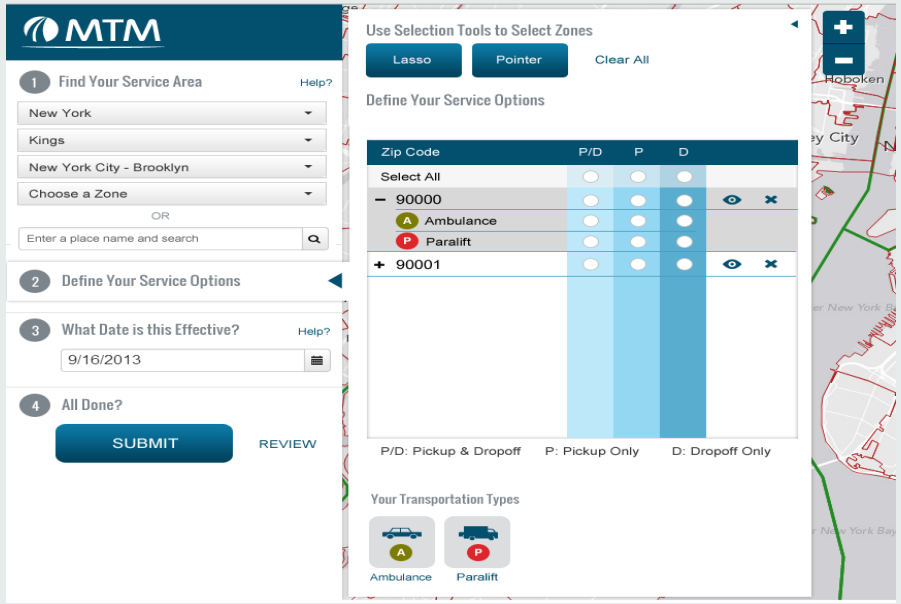


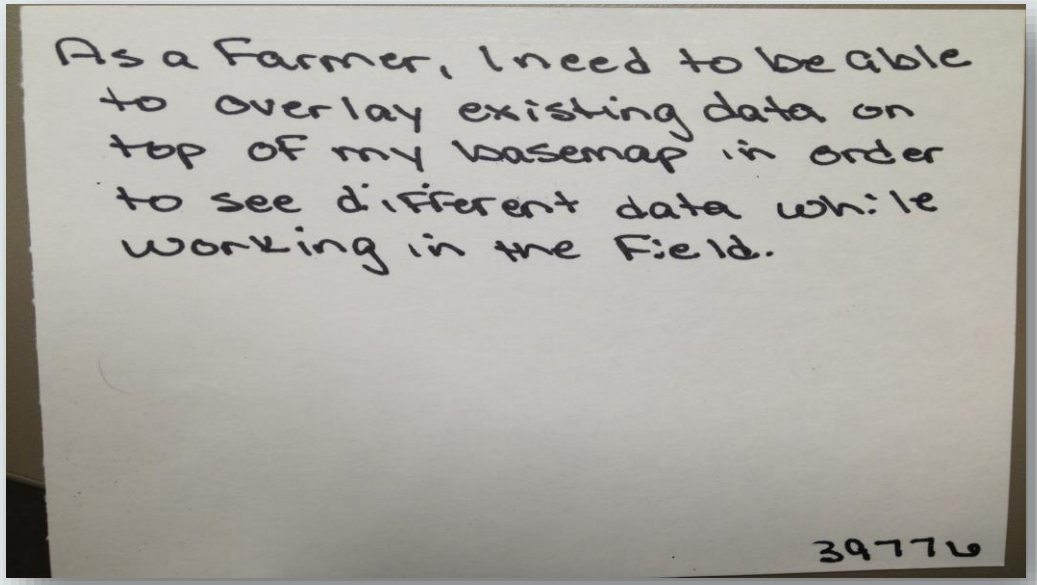
Consider grouping user stories into **themes**

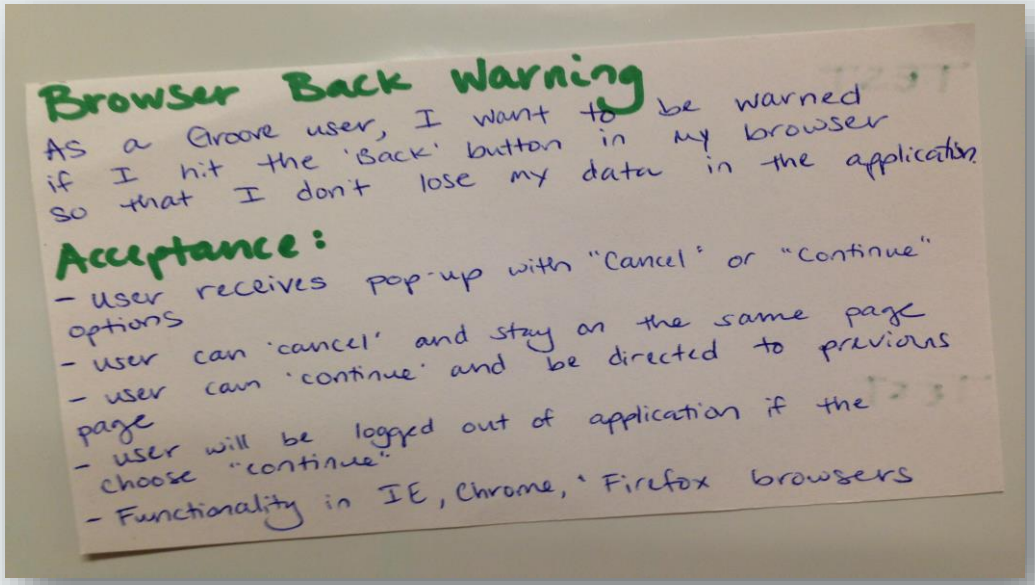
Some things are **not about a [user]**

As a user interface, I need to look like the following image so that I can be intuitive to use









Use **paper cards**

Write out **concise** user stories

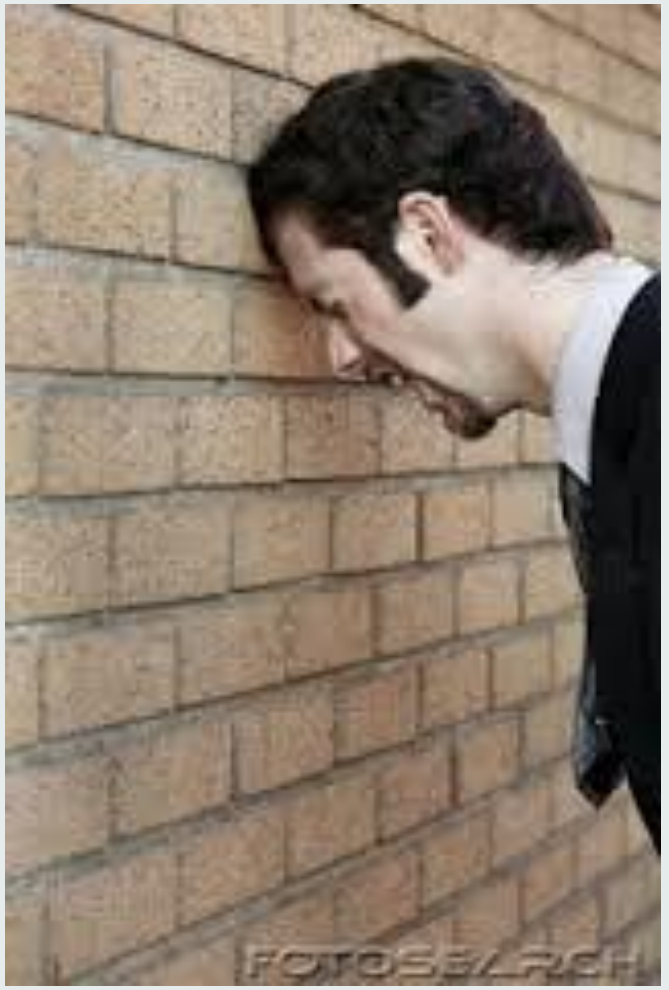
Define **acceptance criteria** for each user story

Keep your stories **visible**

Watch out for the ‘Gotchas’

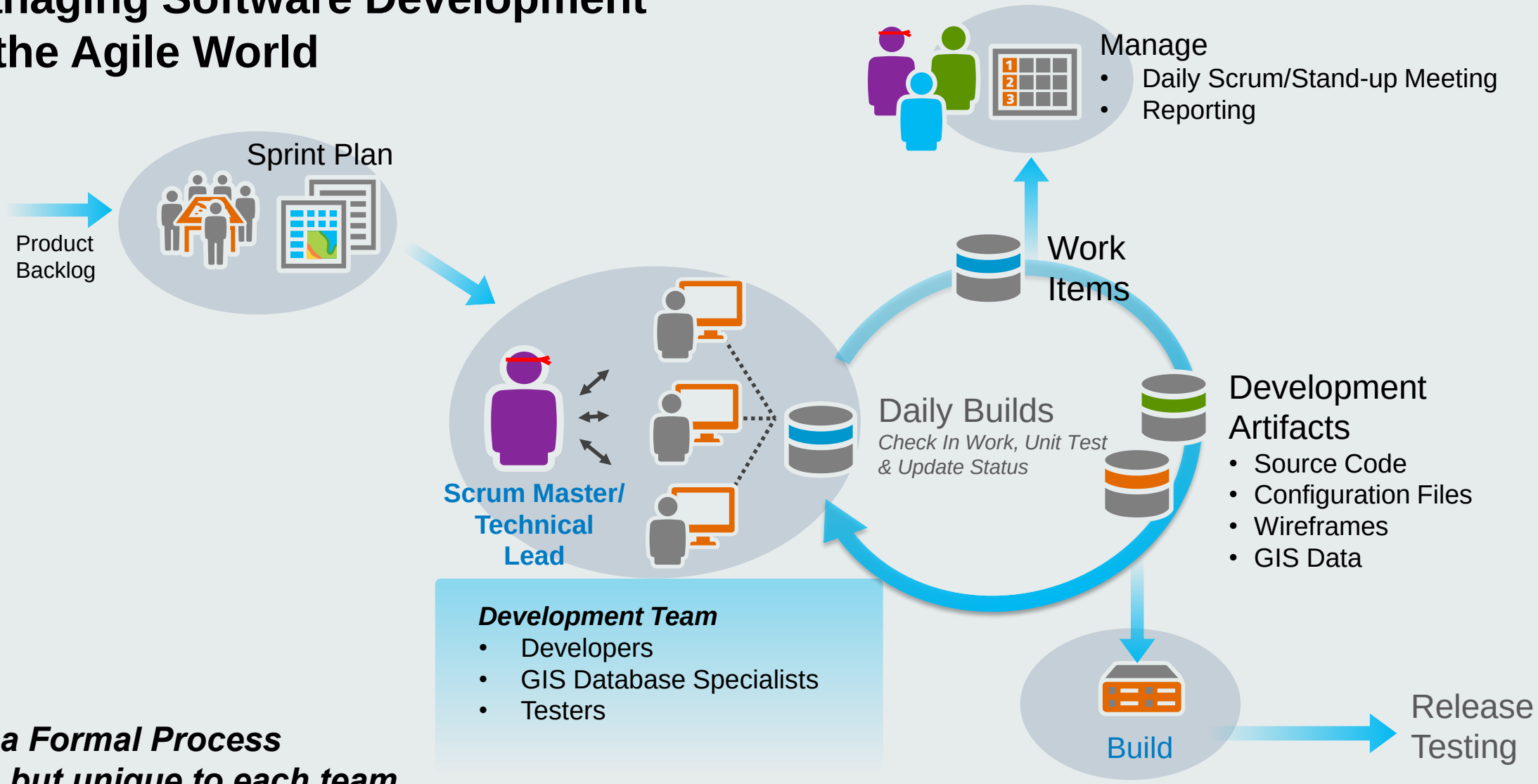
Things to avoid

- Avoid long lists of acceptance criteria on a single user story
- Prepare for conflicting requirements
- Avoid requirements that are ambiguous
- Avoid requirements that describe HOW
- Requirements must have a “reason”
- Avoid moving forward on development until after the customer has reviewed the design
- Don’t forget to prioritize



Exercise: Creating User Stories

Managing Software Development in the Agile World



…a Formal Process
… but unique to each team

Example: Agile Daily Stand Up Meeting for Visibility

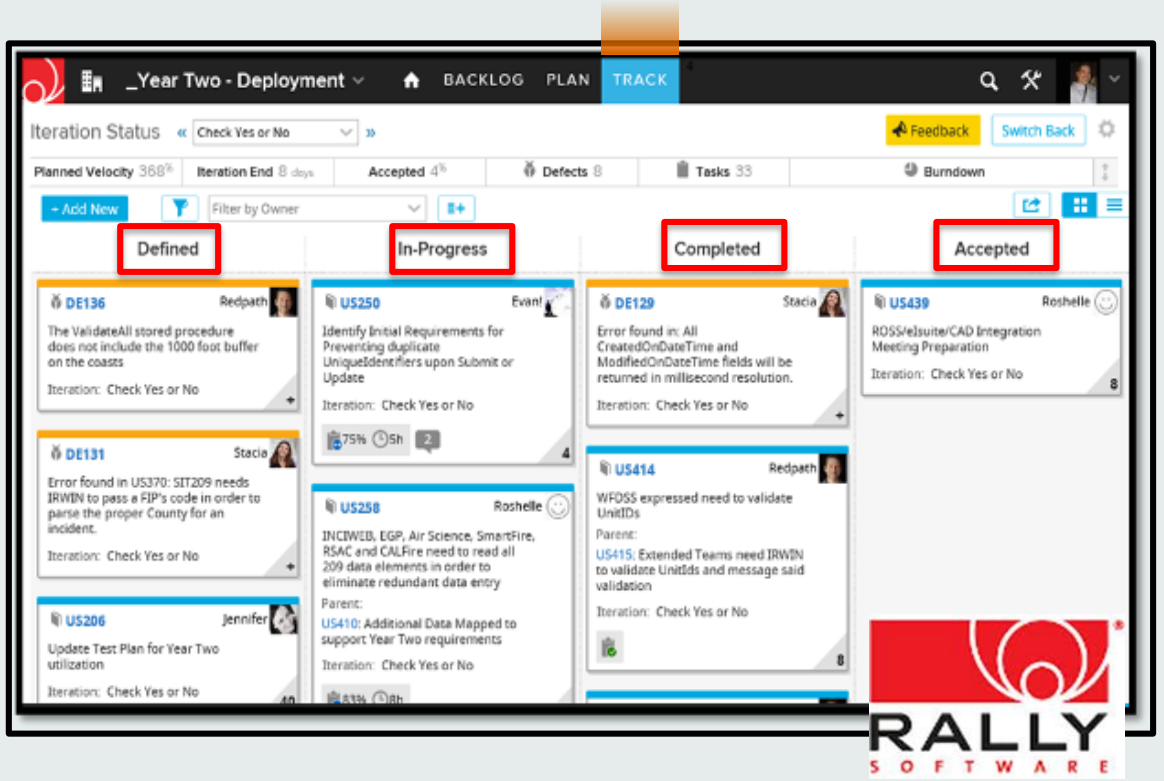
15 Minutes, 3 Important Questions

What did you complete yesterday

What have you planned for today

Are you facing any obstacles?

Defined In Progress Completed Accepted

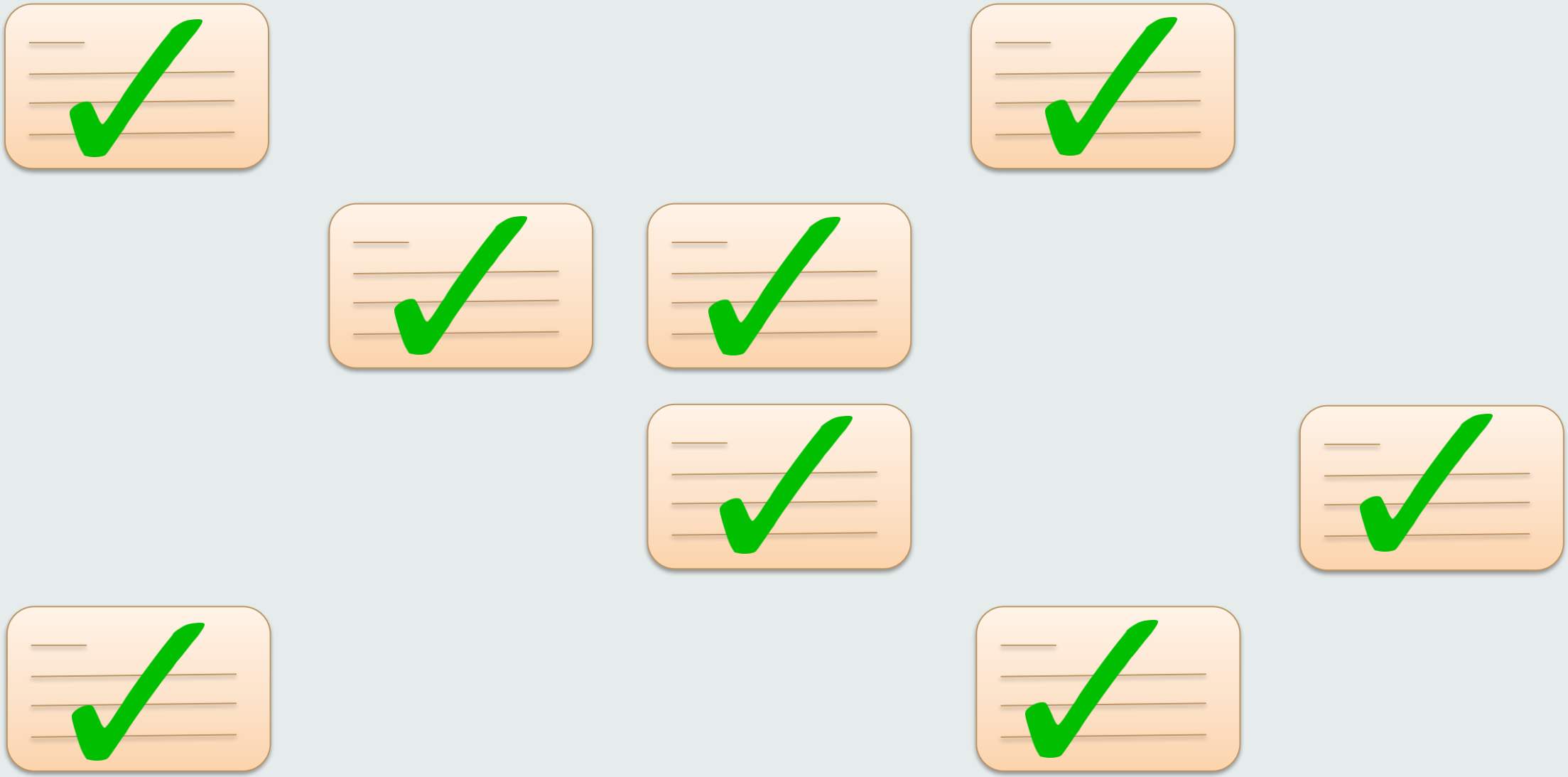


https://www.rallydev.com/

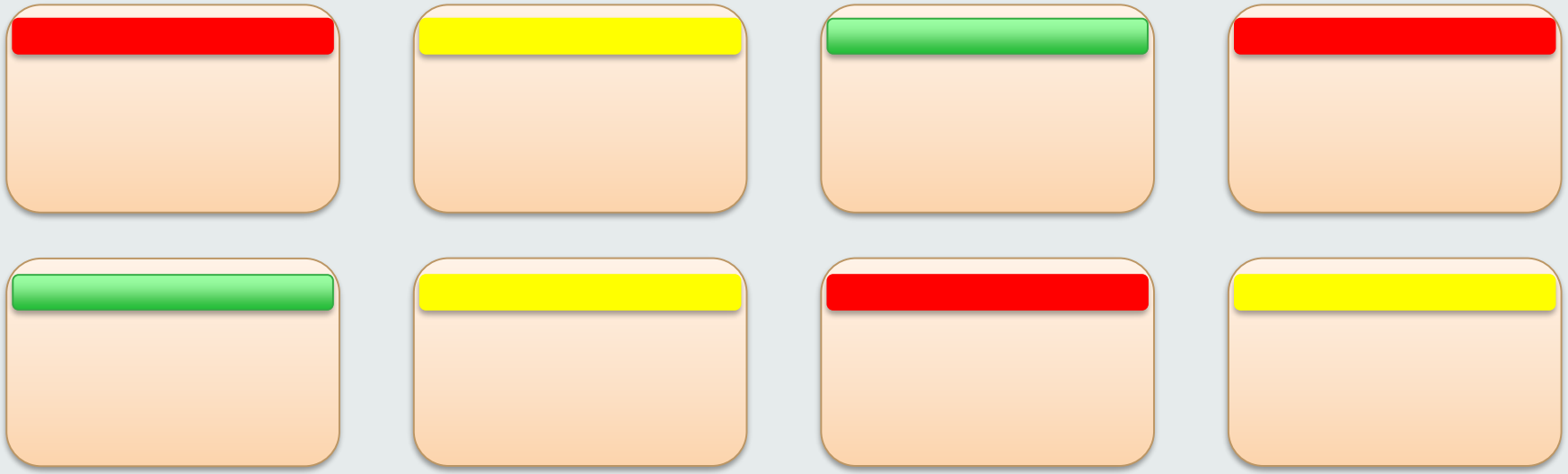
Product Backlog

Wish List

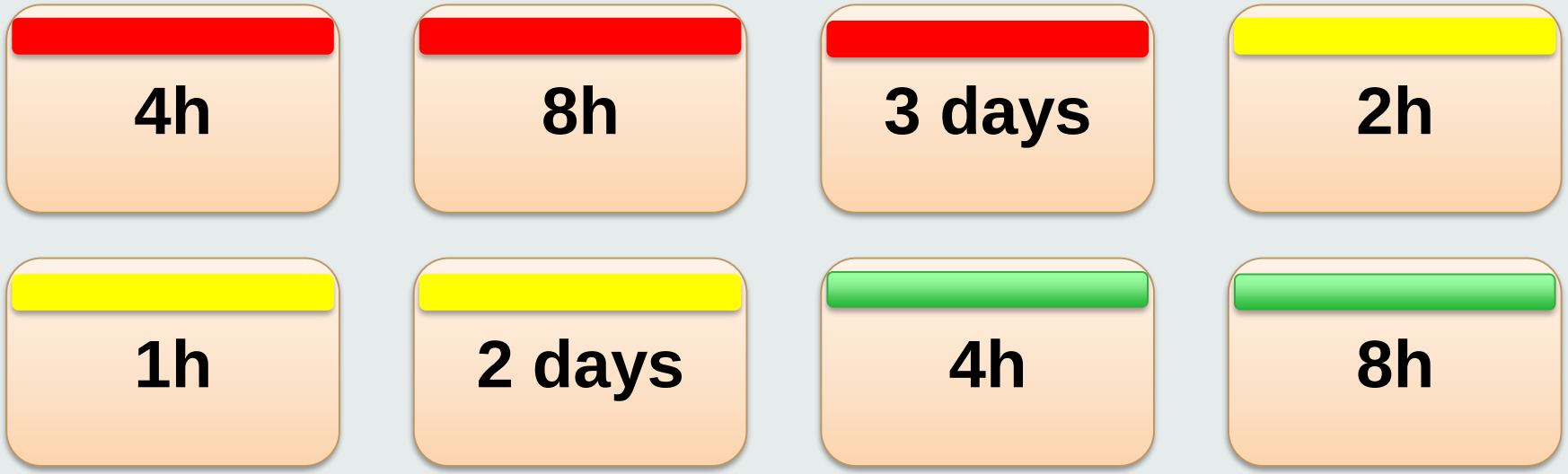
Sprint Backlog



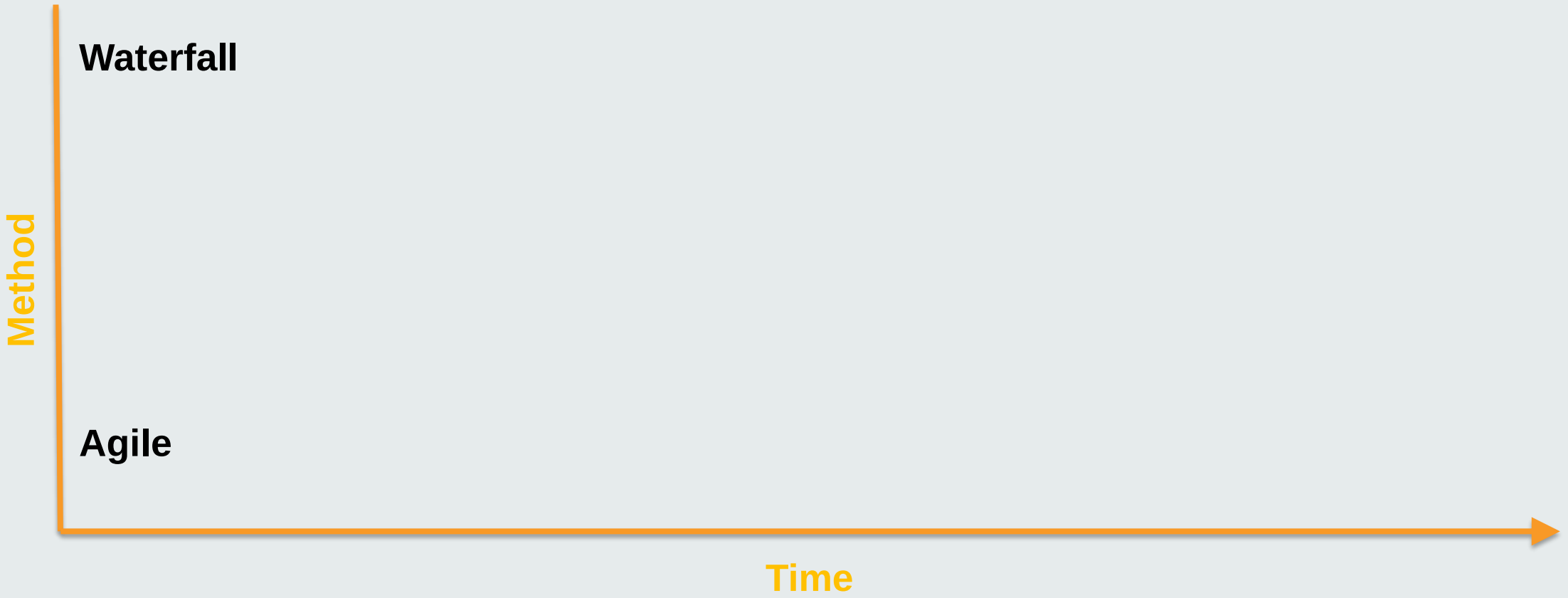
Sprint Backlog



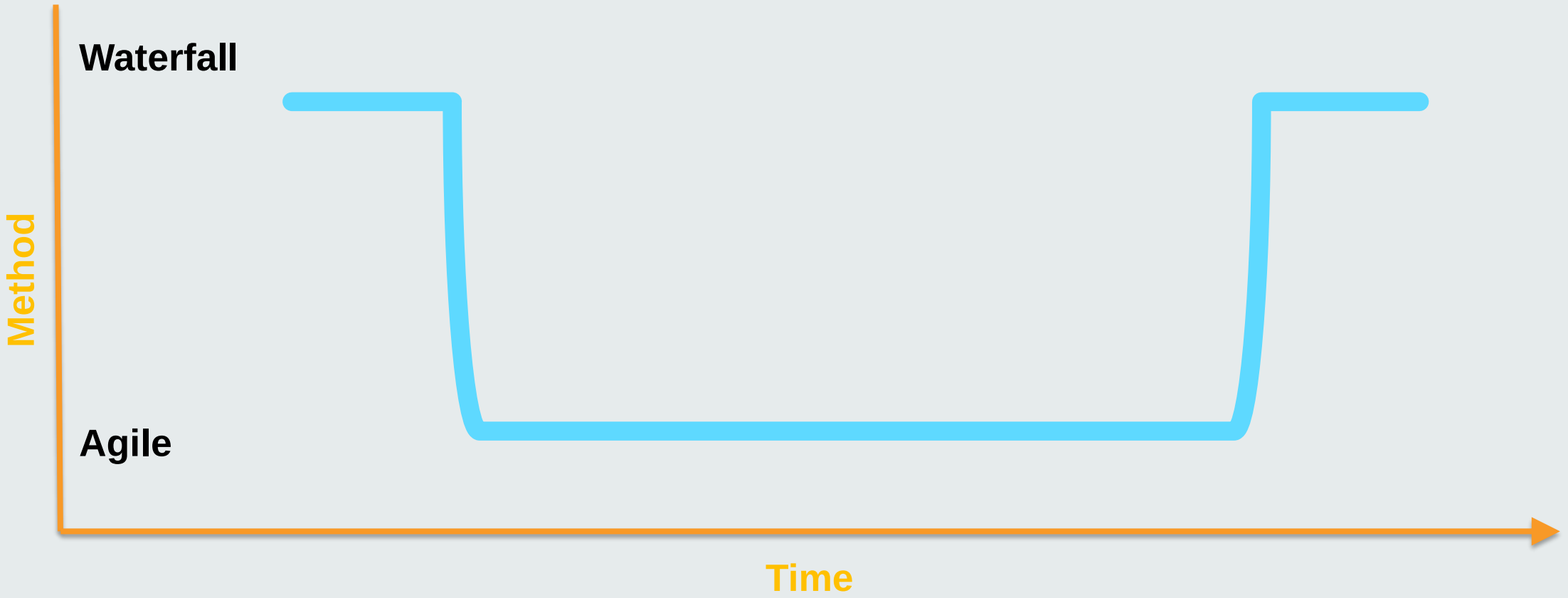
Sprint Backlog



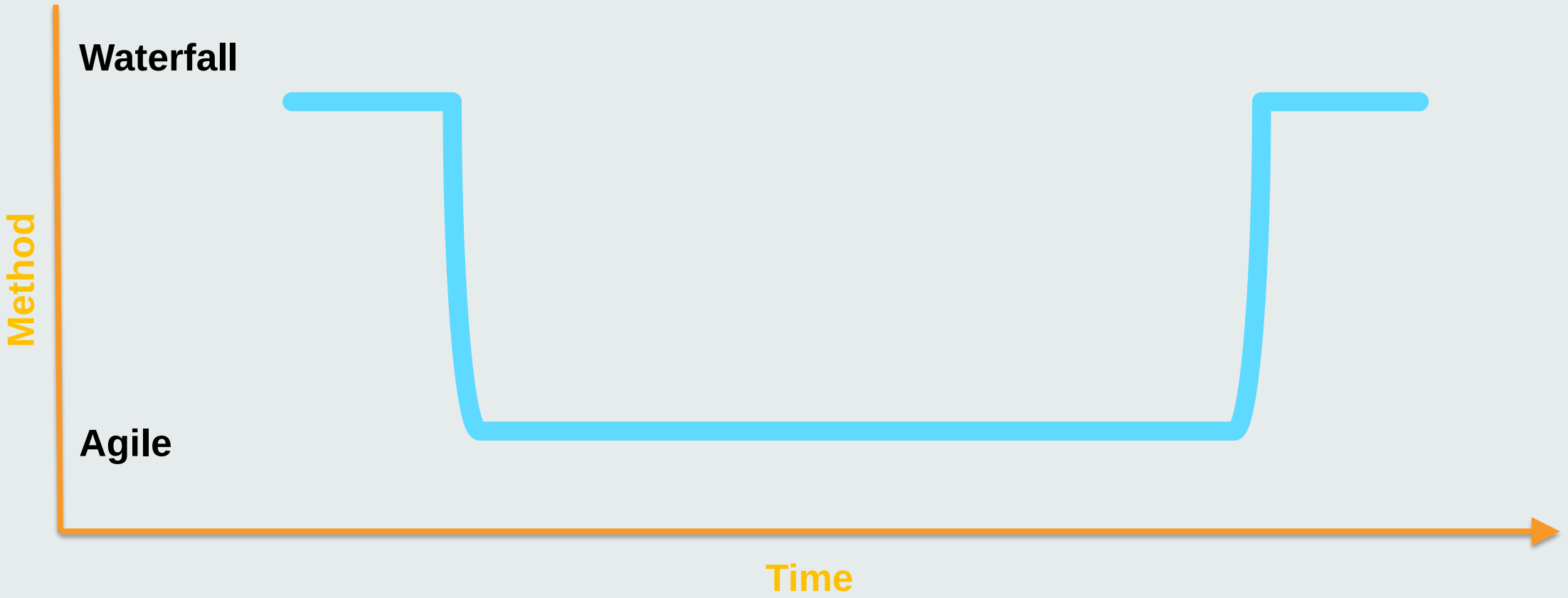
Using Agile in a Consulting Project



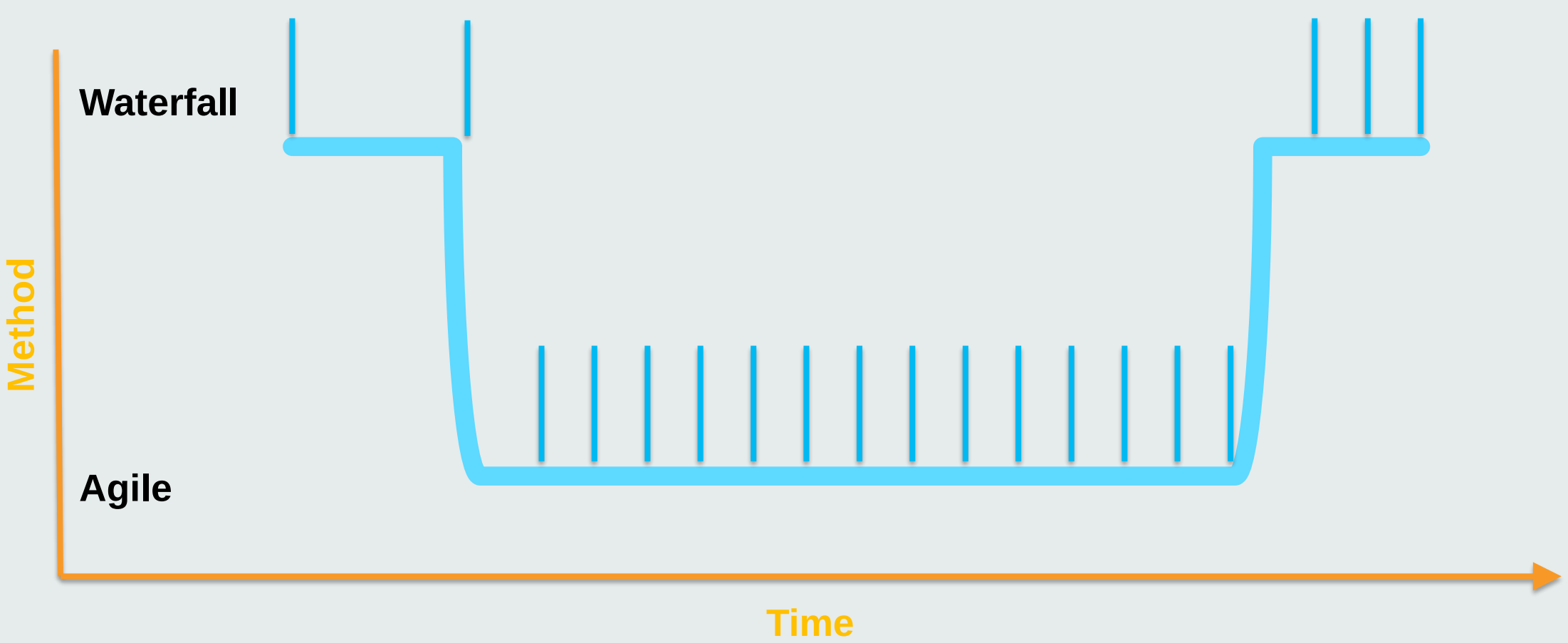
Using Agile in a Consulting Project



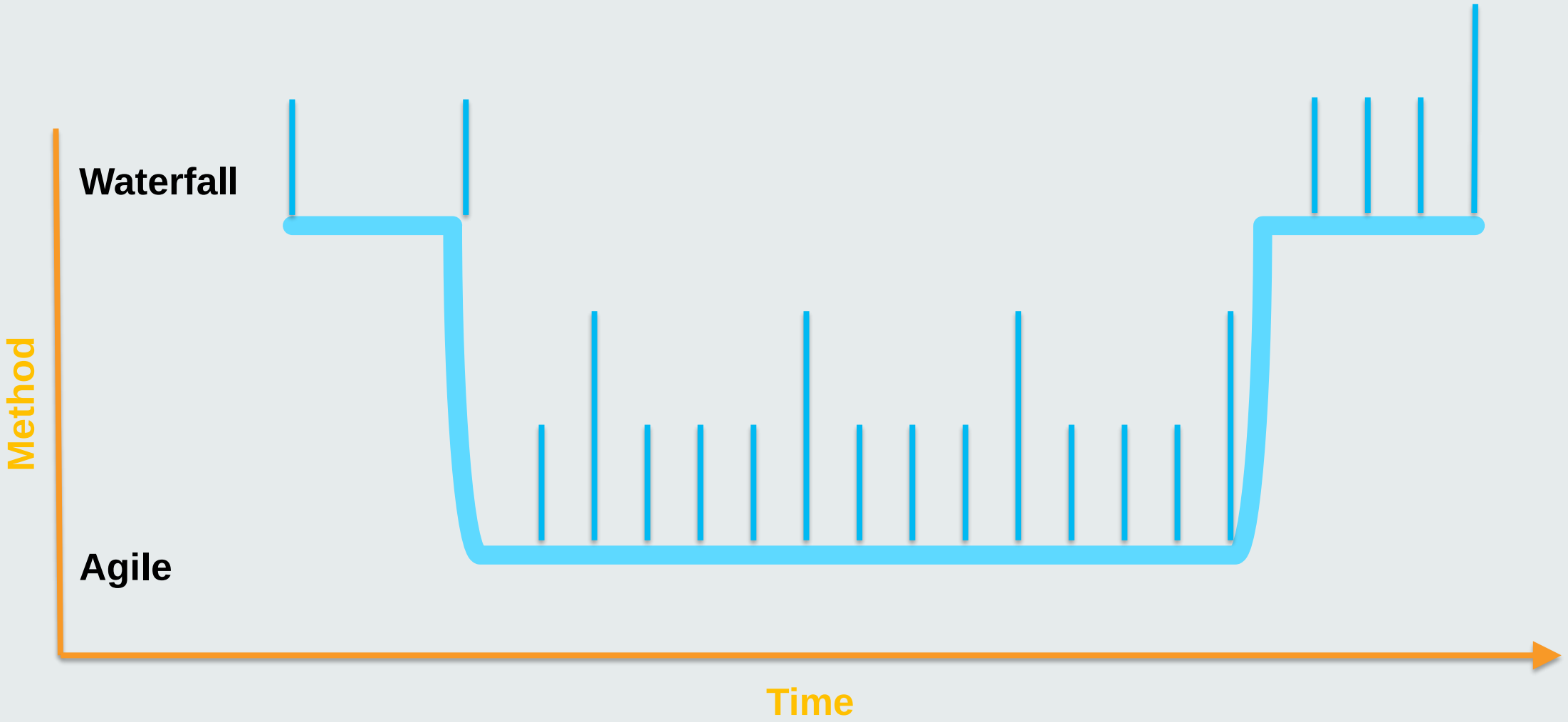
Using Agile in a Consulting Project



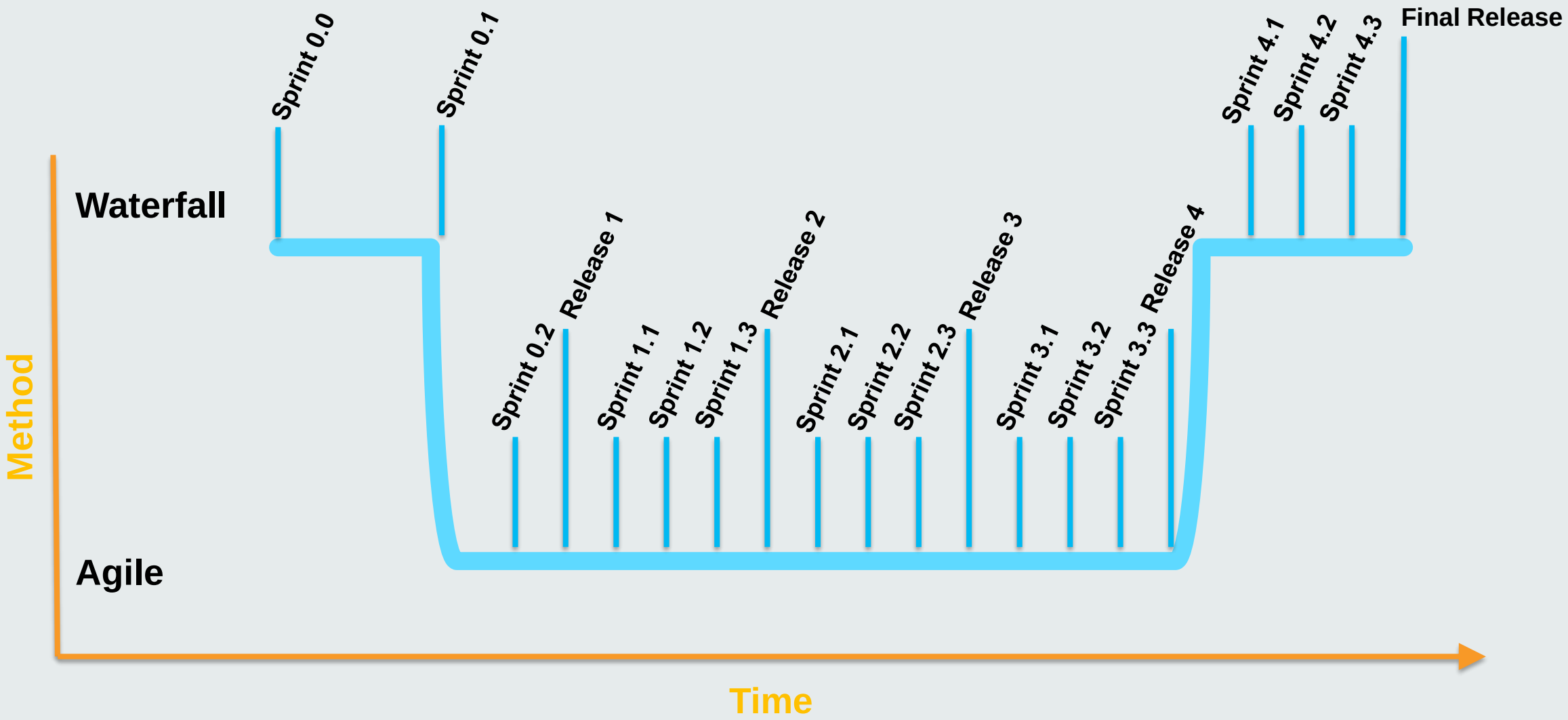
Using Agile in a Consulting Project



Using Agile in a Consulting Project



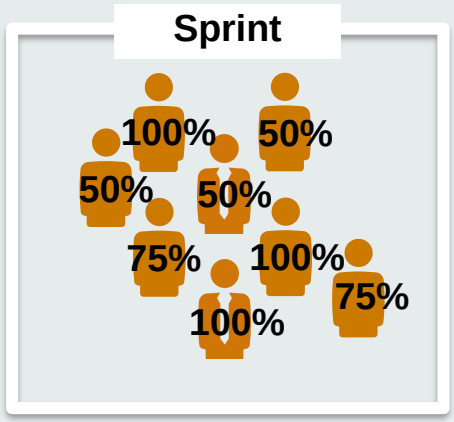
Using Agile in a Consulting Project



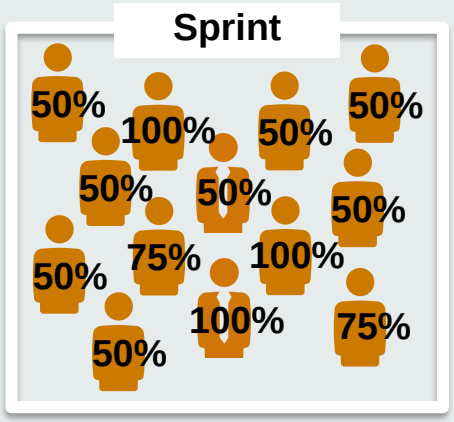
Managing Resources

Plan A

Plan B



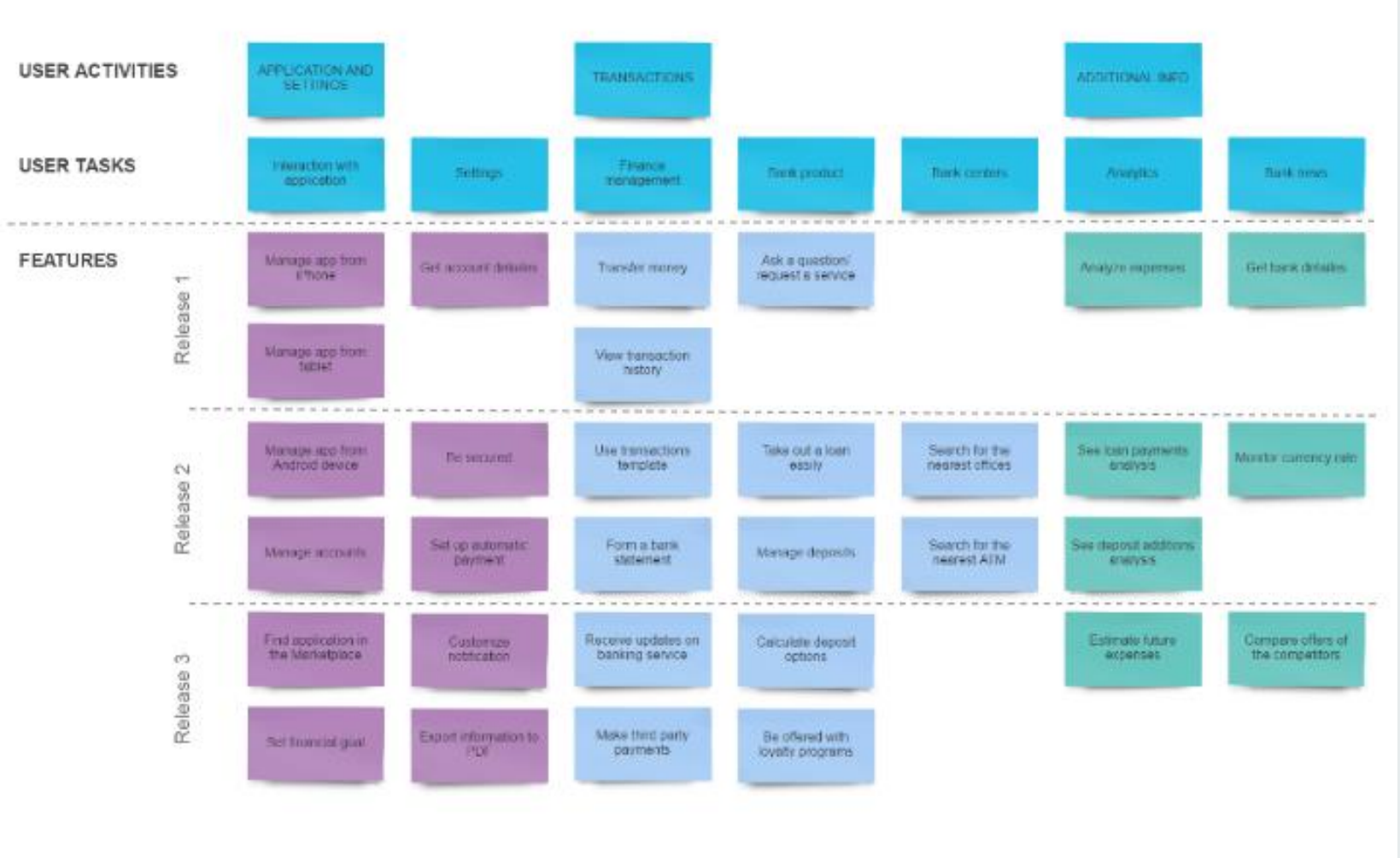
Plan Z



Tools

Tool to assist in Release/Sprint Planning

RealTimeBoard

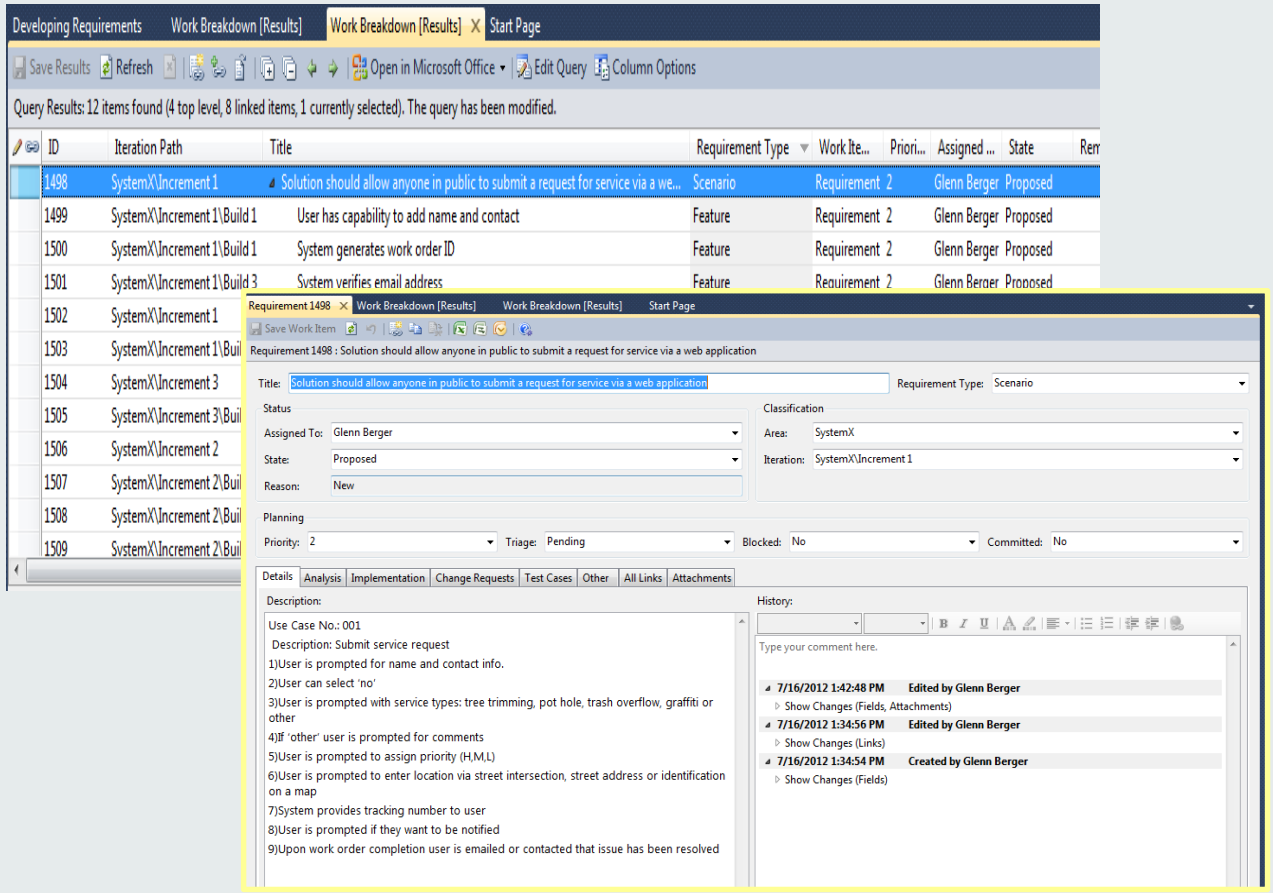


Use Tools to Manage Requirements

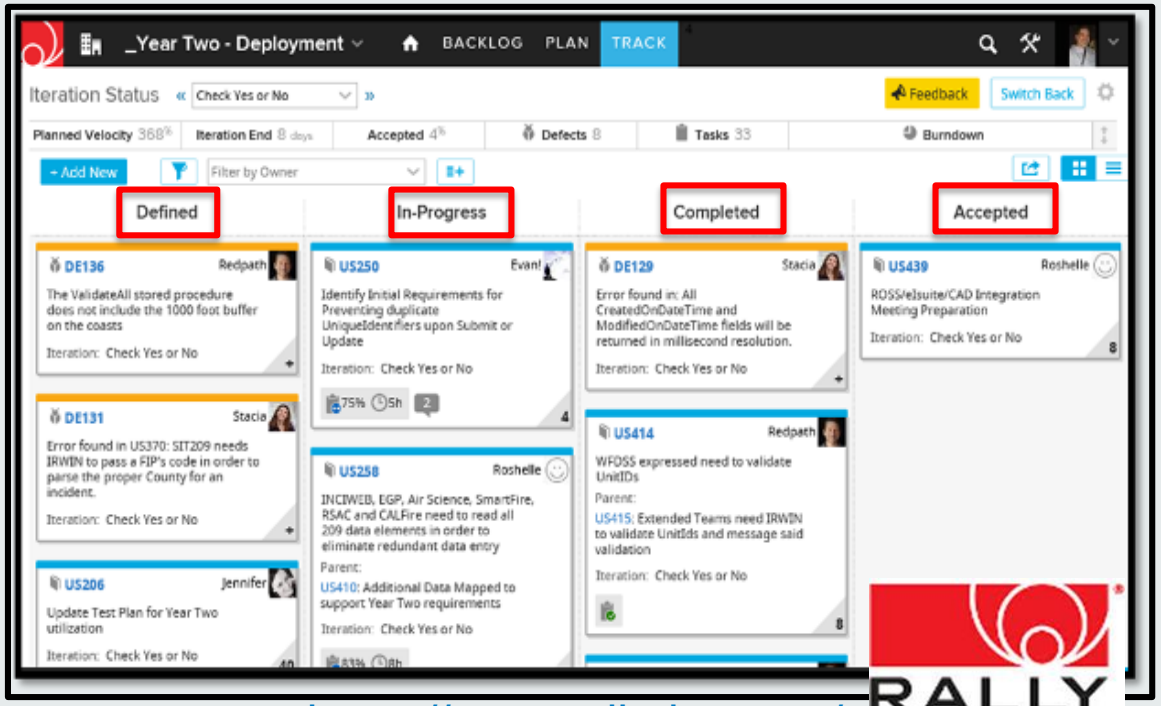
Appendix A
Afternoon
Page 18

Requirements Traceability Matrix

Microsoft Team Foundation Server (TFS)

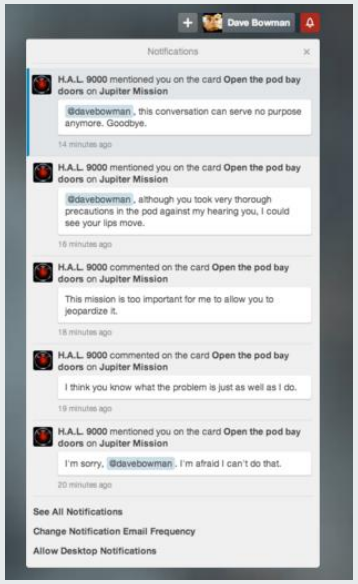


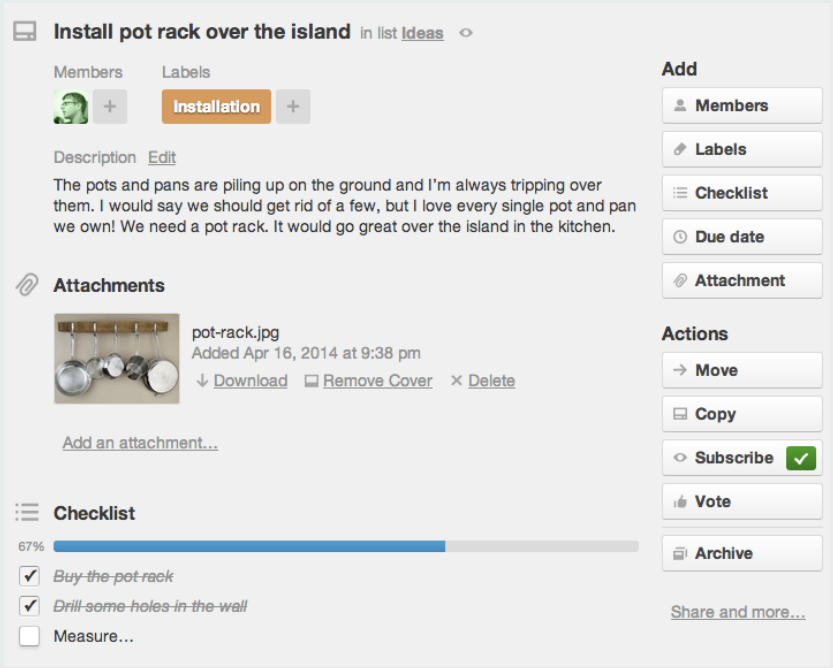
Rally Agile Implementation Tool



https://www.rallydev.com/

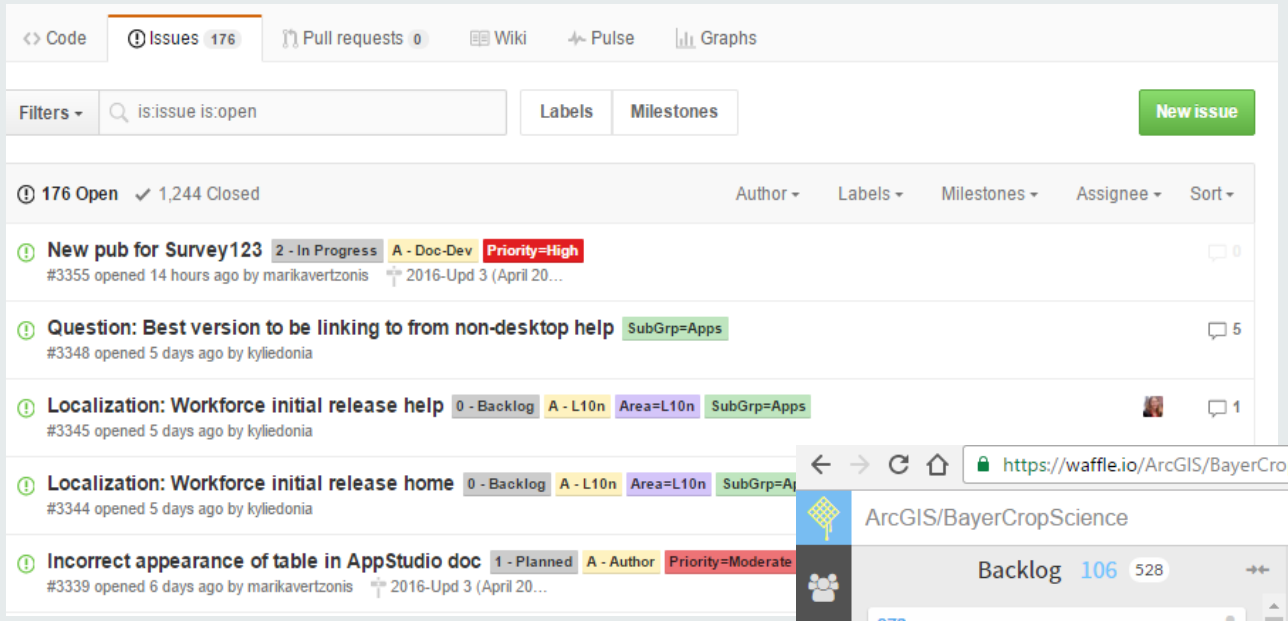
Using Trello

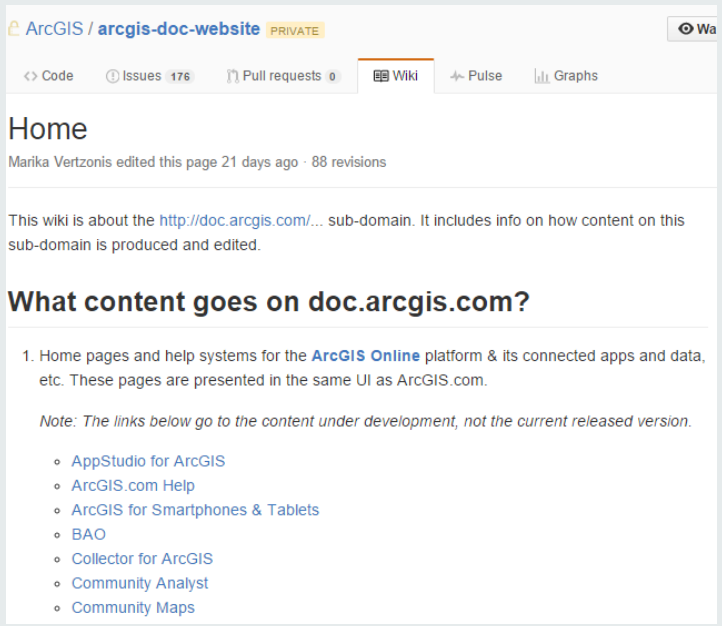


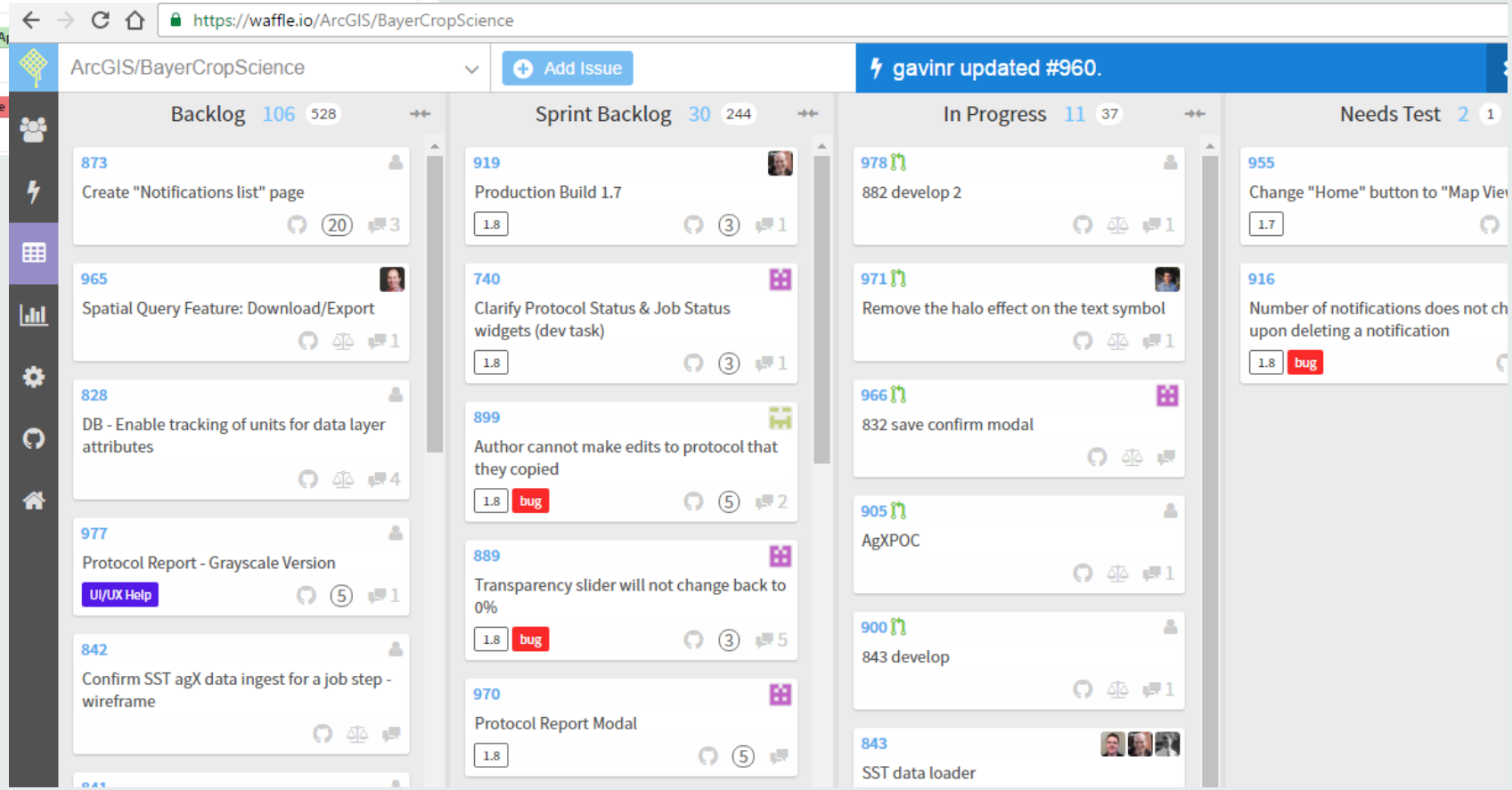


https://www.rallydev.com/

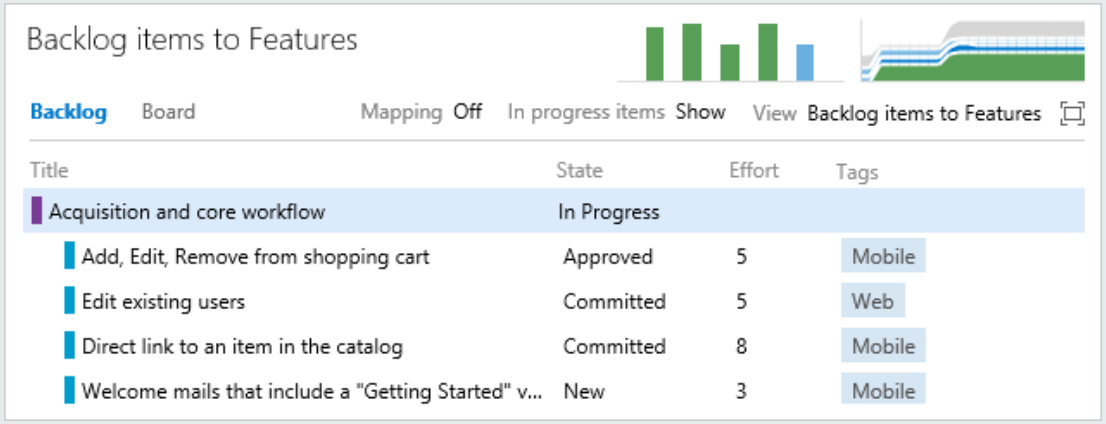
Using GitHub

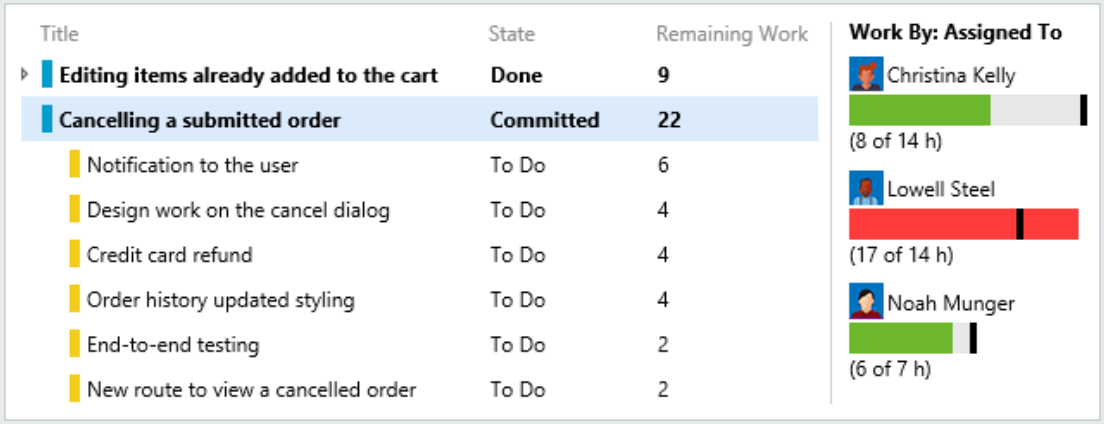


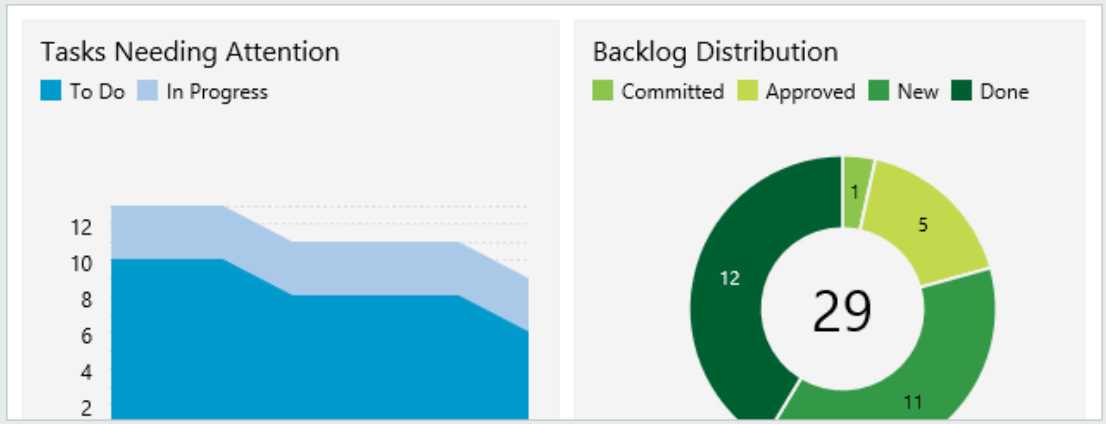


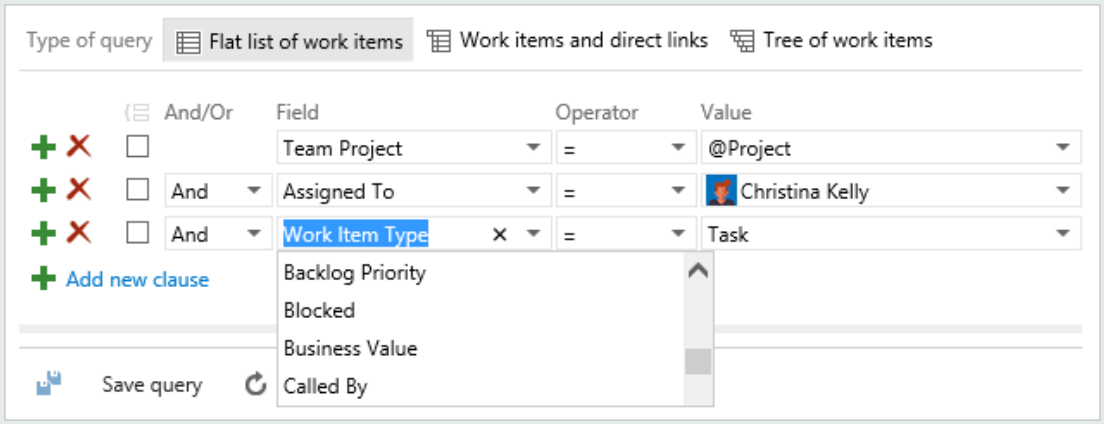


Using TFS


Backlog items to Features
Backlog
Board
Mapping Off
In progress items Show
View Backlog items to Features
Title
State
Effort
Tags
Acquisition and core workflow
In Progress
Add, Edit, Remove from shopping cart
Approved
5
Mobile
Edit existing users
Committed
5
Web
Direct link to an item in the catalog
Committed
8
Mobile
Welcome mails that include a "Getting Started" v...
New
3
Mobile


Title
State
Remaining Work
Editing items already added to the cart
Done
9
Cancelling a submitted order
Committed
22
Notification to the user
To Do
6
Design work on the cancel dialog
To Do
4
Credit card refund
To Do
4
Order history updated styling
To Do
4
End-to-end testing
To Do
2
New route to view a cancelled order
To Do
2
Work By: Assigned To
Christina Kelly
(8 of 14 h)
Lowell Steel
(17 of 14 h)
Noah Munger
(6 of 7 h)


Tasks Needing Attention
To Do
In Progress
12
10
8
6
4
2
Backlog Distribution
Committed
Approved
New
Done
1
5
12
29
11


Type of query
Flat list of work items
Work items and direct links
Tree of work items
And/Or
Field
Operator
Value
Team Project
=
@Project
And
Assigned To
=
Christina Kelly
And
Work Item Type
=
Task
Add new clause
Backlog Priority
Blocked
Business Value
Called By
Save query

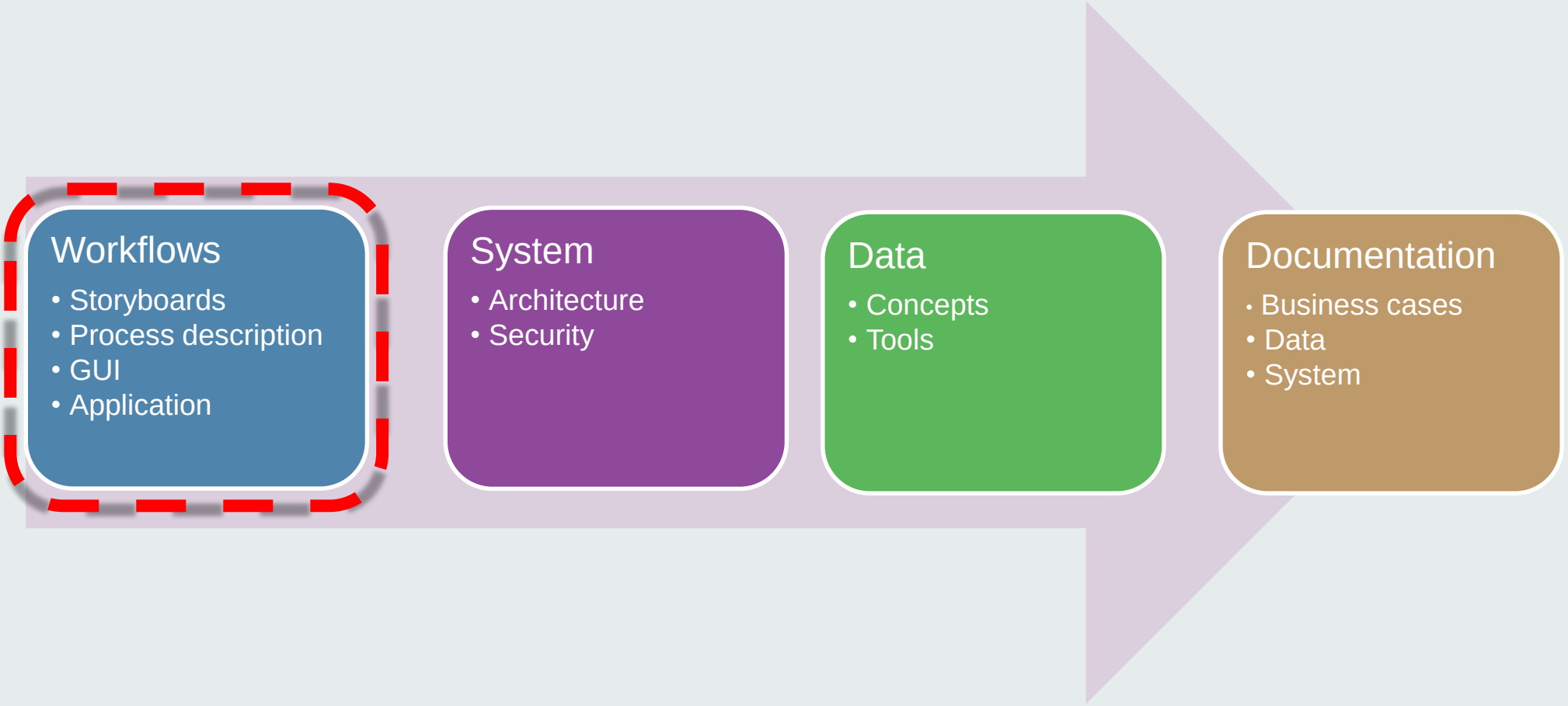
Making a Decision

Project Considerations	Trello	GitHub	TFS
Requirements are Proprietary			
Mobile App			
Easy to setup			
Estimation tools			
Scheduling tools			
Automated Burndown chart			
Easily integrated with Visual Studio for Code Repository			
Capacity Planning			
Exports to MPP and Excel			

Avoiding Design Nightmares

https://www.youtube.com/watch?v=BKorP55Aqvg

Design


Workflows
• Storyboards
• Process description
• GUI
• Application
System
• Architecture
• Security
Data
• Concepts
• Tools
Documentation
• Business cases
• Data
• System

What needs to get designed?

The Front

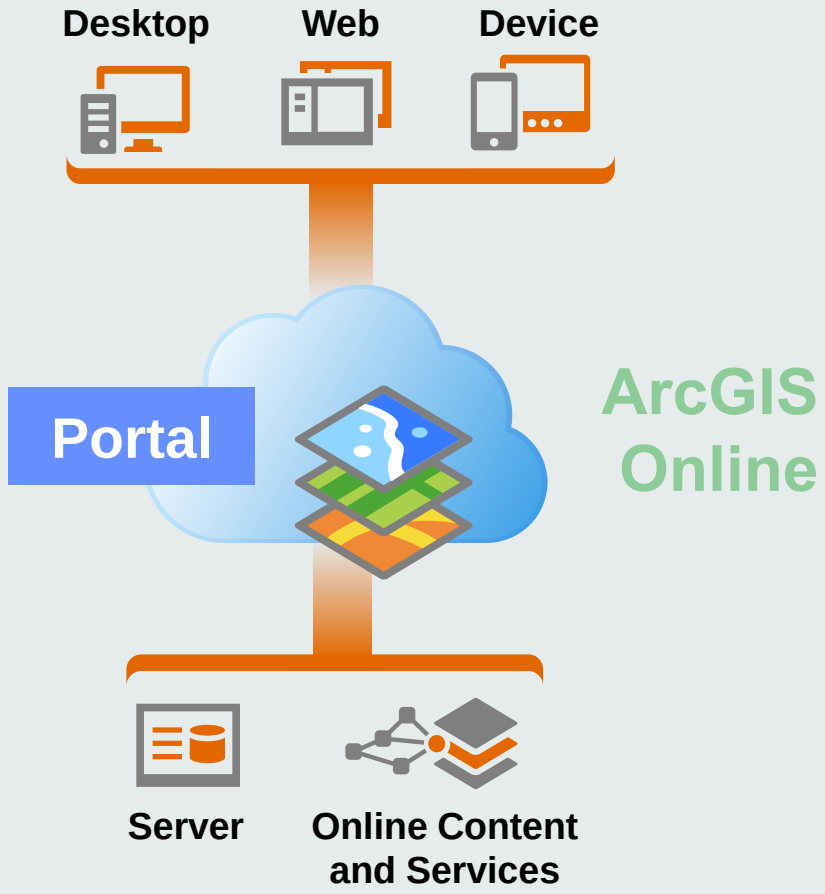
- User Experience
- Information Products (Maps, Reports, Models)

The Middle

- Business Logic
- Integration Points (Between components, to enterprise systems)

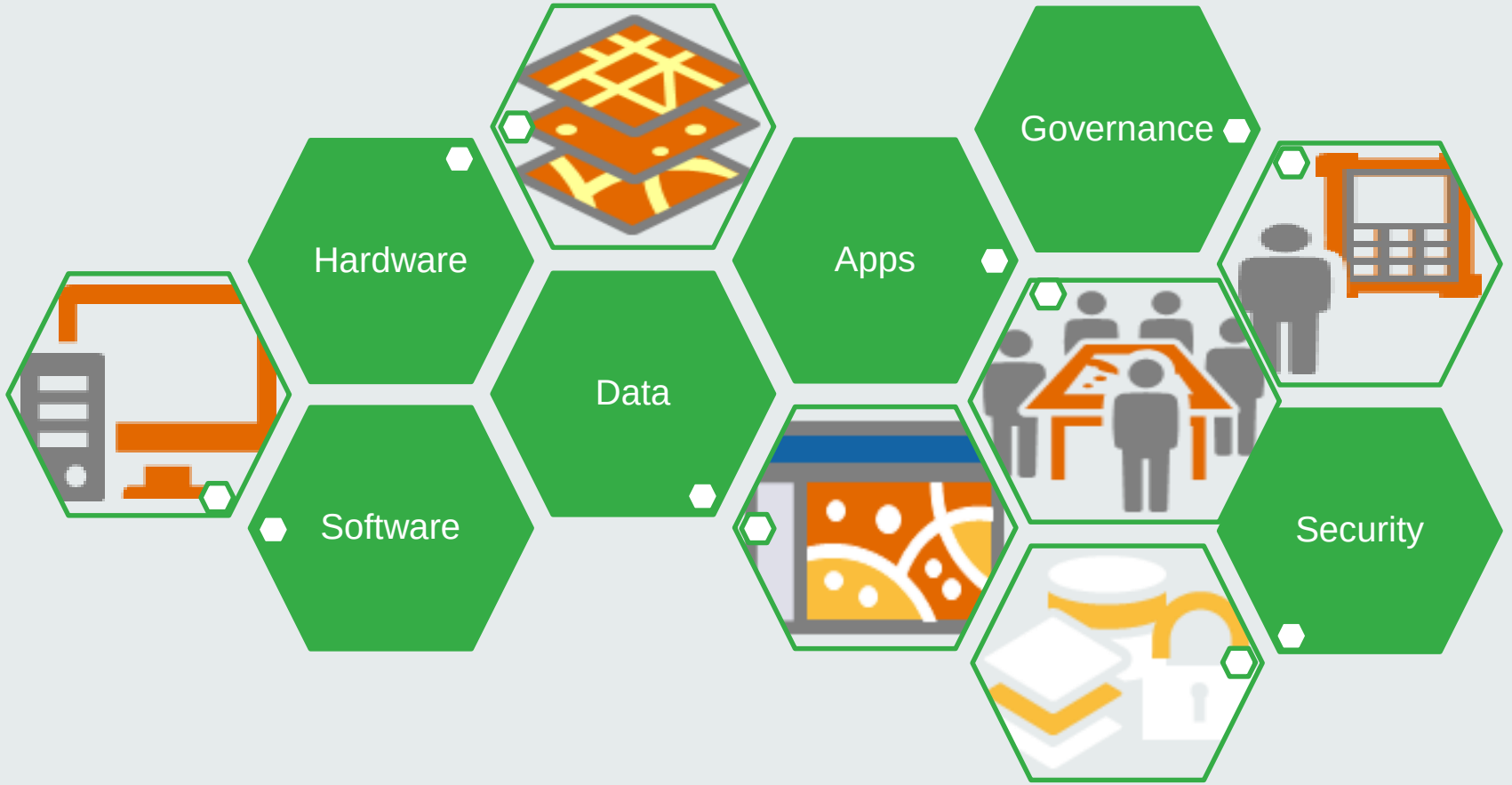
The Back

- Databases, content, services
- Architecture (→ Foundational Decisions)



Design Components Together

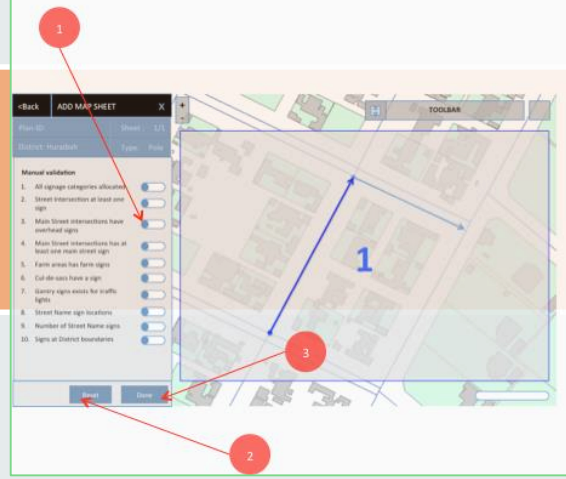
Interoperable



Develop mockups of the UI and work process

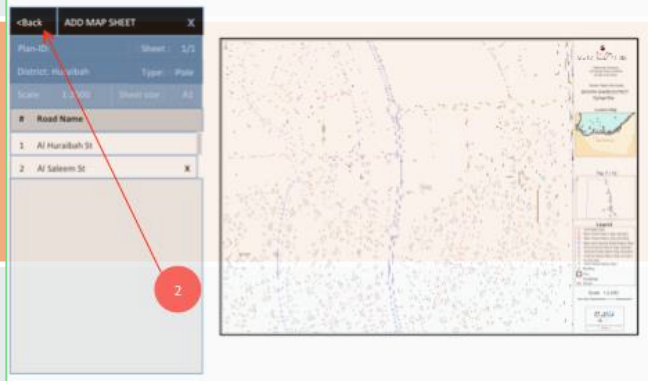
- **Convert user stories into storyboards visually describing the workflow process**
- **A picture is worth more than a 100 words**
- **Describe user and system actions for each storyboard frame**

Define Signage Plan – Validate sheet form



1. User sets Manual Validation buttons
   1. System displays button settings
2. User selects 'Reset'
   1. System resets all validation buttons
3. User selects 'Done'
   1. System writes to Journal
   2. System returns to Add Sheet Form
   3. TBD: System stores validation settings

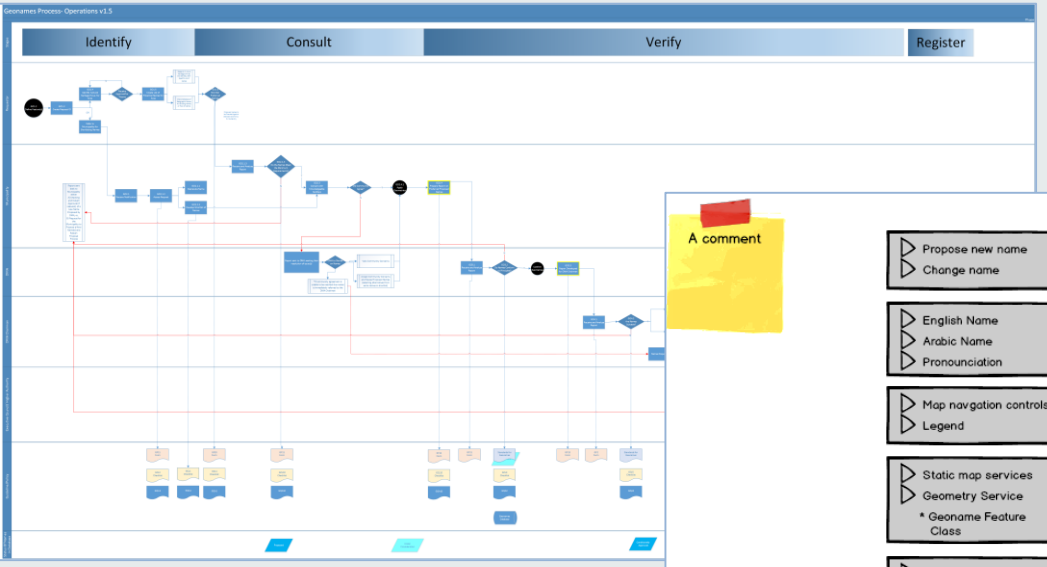
Define Signage Plan – View Sheet Form

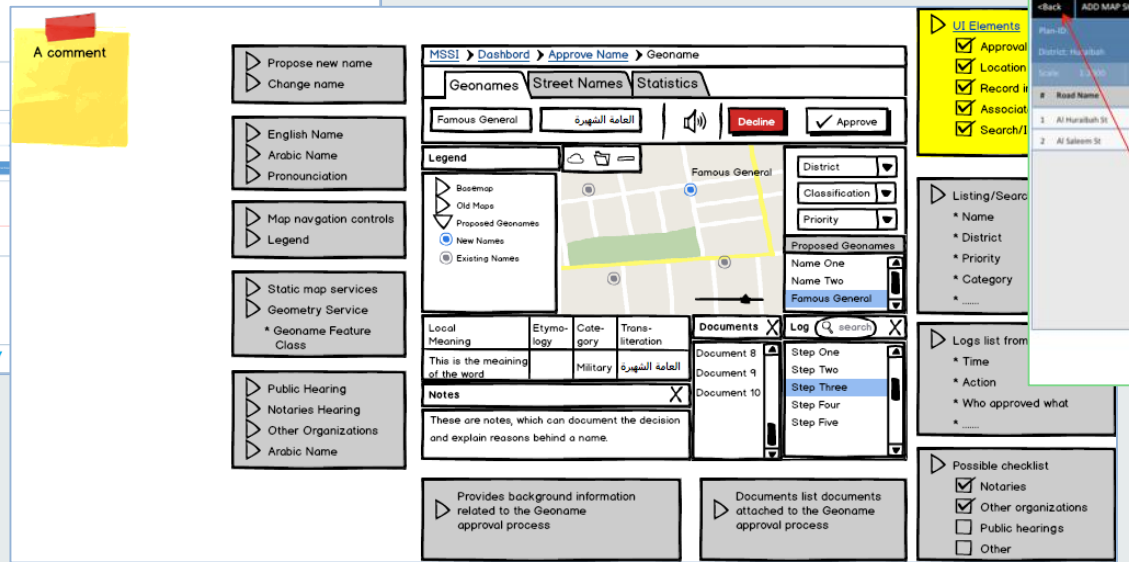


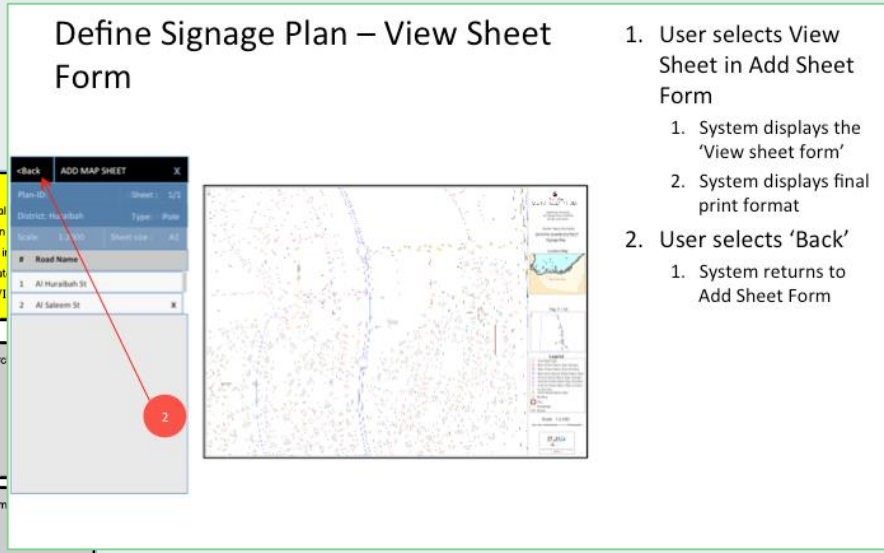
1. User selects View Sheet in Add Sheet Form
   1. System displays the 'View sheet form'
   2. System displays final print format
2. User selects 'Back'
   1. System returns to Add Sheet Form

Develop mockups of the UI and work process

- **Available tools**
  - **Visio**
  - **Balsamiq**
  - **Power Point / Keynote**
  - **Photo software – PSP/Corel…**







Process descriptions

- **Process identification**
- **Actions by the user**
  - **Inputs**
  - **Format**
  - **Limitations**
- **System action**
  - **System process description – properties**
  - **Calculation formula's**
  - **Rules for processing**
  - **Data and domain values**
  - **Test cases with input values and results**

Application Design Themes

- **Immediate capabilities**
- **Simple, focused apps**
- **Traceability to requirements**
- **Engage the customer**
- **Involve developers and end users**
- **Focus on mitigating risk**
- **Prototype**

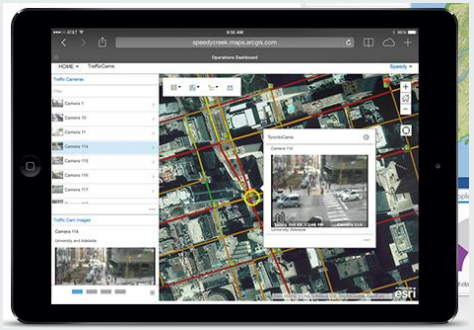
Use ArcGIS Online and Resource Centers

http://www.arcgis.com/features/
http://resources.arcgis.com/en/home/
http://github.com/esri

Example Templates for Water Leak Management

Web Isolation Trace

Operations Dashboard for ArcGIS

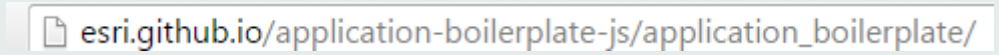


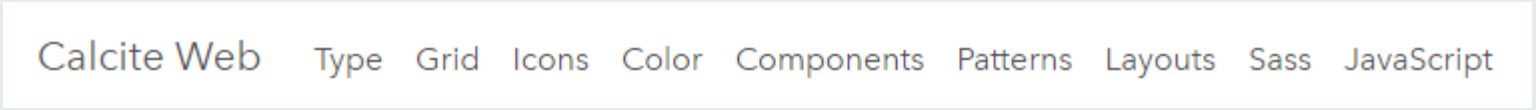
Mobile Leak Inspector

Field Asset Inspections

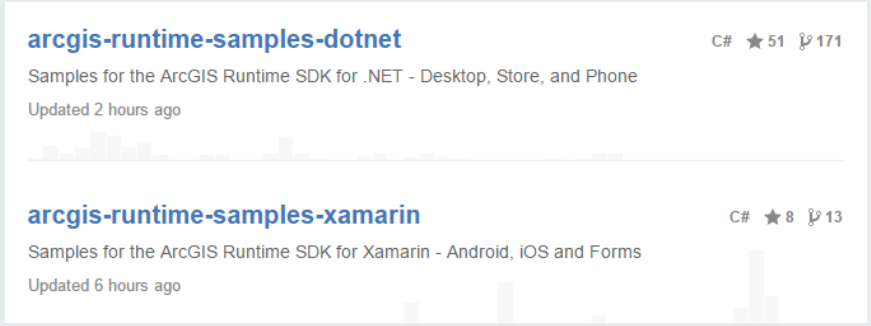
Application Design Tools, Technology

- **Scaffolding tools such as Yeoman**
- **Existing application templates published by Esri and others**
- **Existing styling frameworks such as Bootstrap may guide design**
- **Rebuild from existing templates**



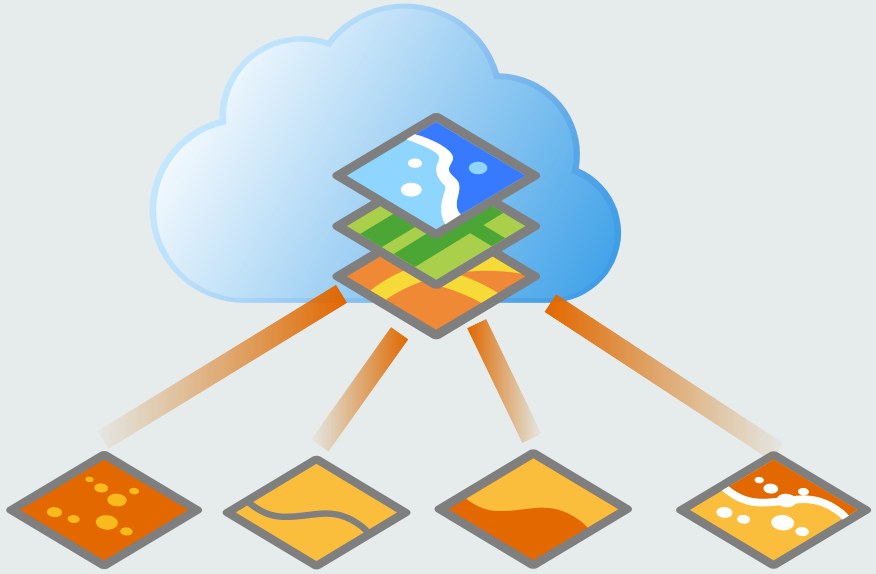


Calcite Bootstrap



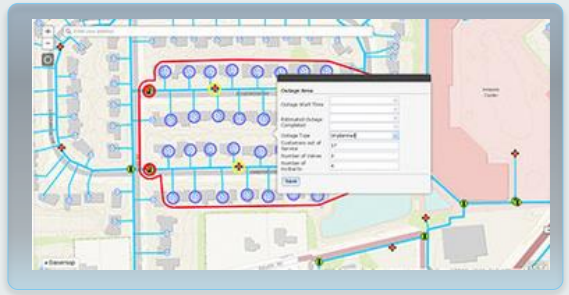
System: Application Design

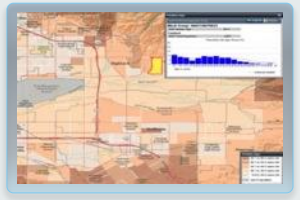
Design Information Products

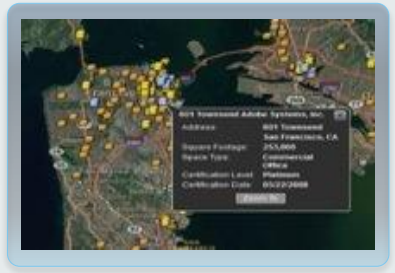


- ✓ Consider what's available already
- ✓ Consider your deployment platform!

Maps, reports, models



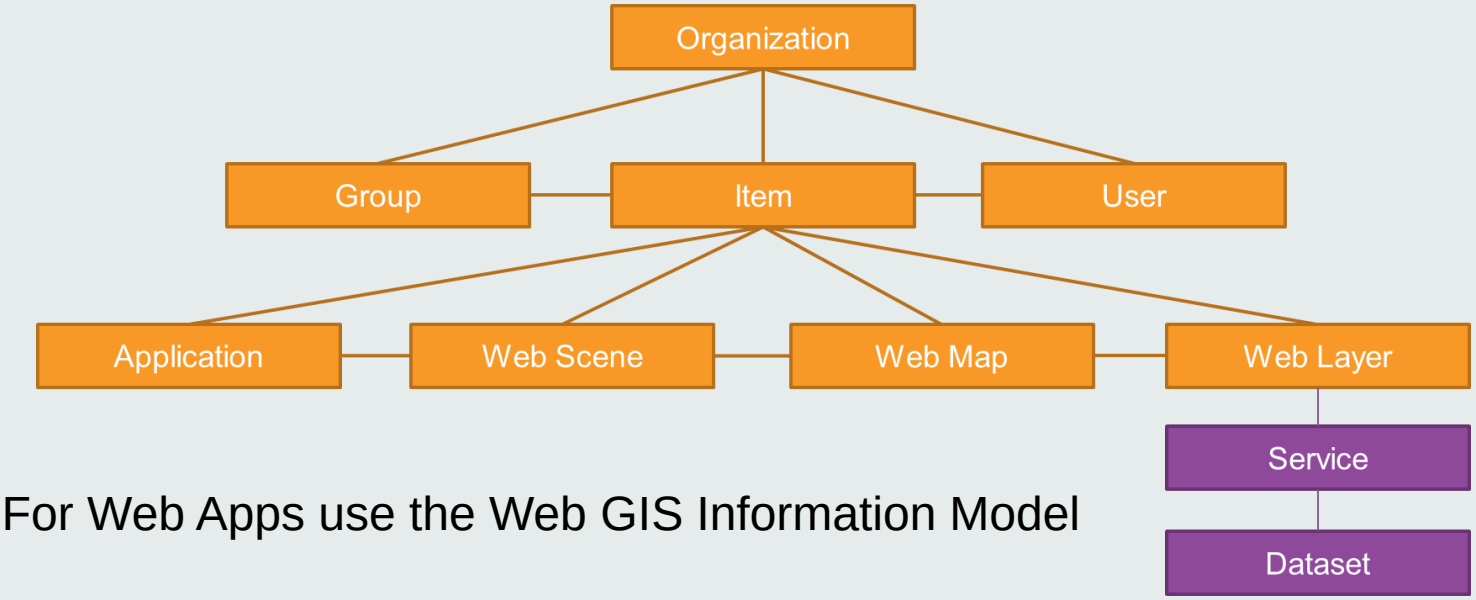




For Web/mobile apps… Design maps as Web Maps

Design Access to Content

- **Applications → Web services → Data**
  - **Web/Mobile – no direct database access**
  - **Requires deliberate & thoughtful authoring**
  - **Different security model**

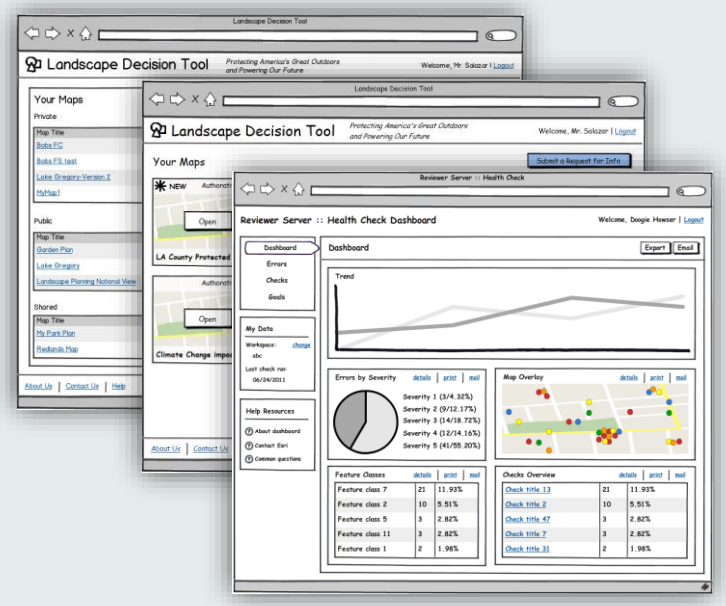


For Web Apps use the Web GIS Information Model

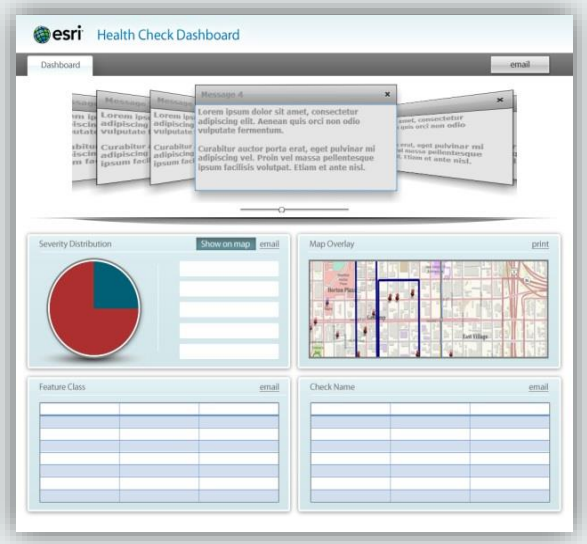
System: Application Design

Design the User Experience – How to Evolve an Interface

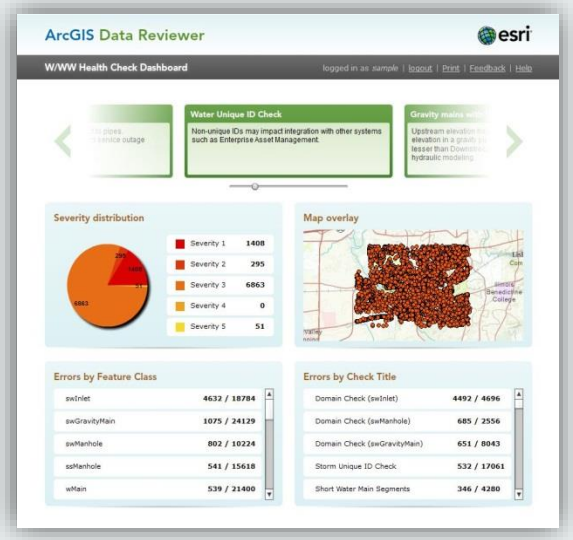
- Wireframes (sketch ideas and workflows, validate assumptions)
- Review Wireframes and conduct Usability Testing
- Look and Feel of the GUI



Wireframes



Prototype

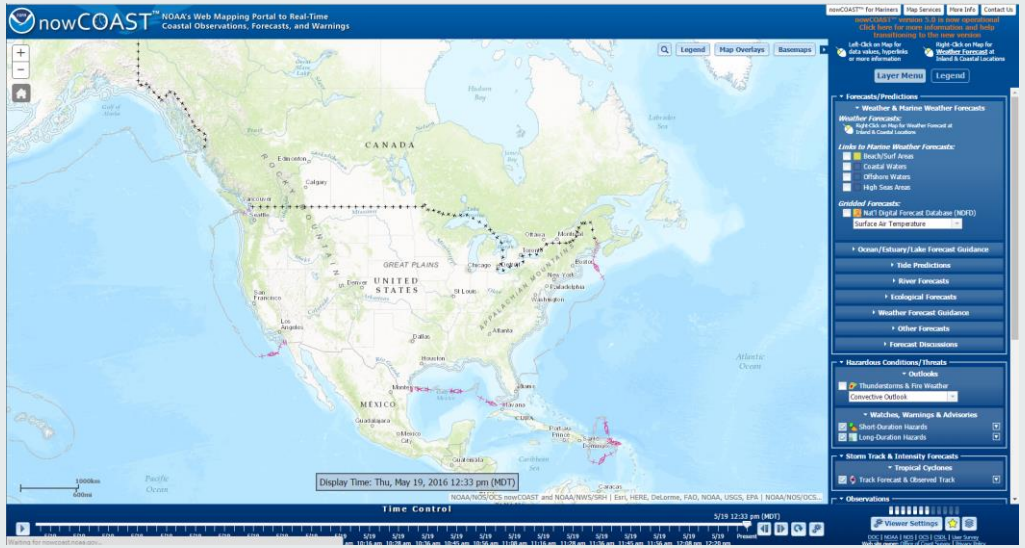


Final GUI

System: Application Design

Design the UX for the common user

- Avoid adding superfluous features
- Easier to add in the future than remove

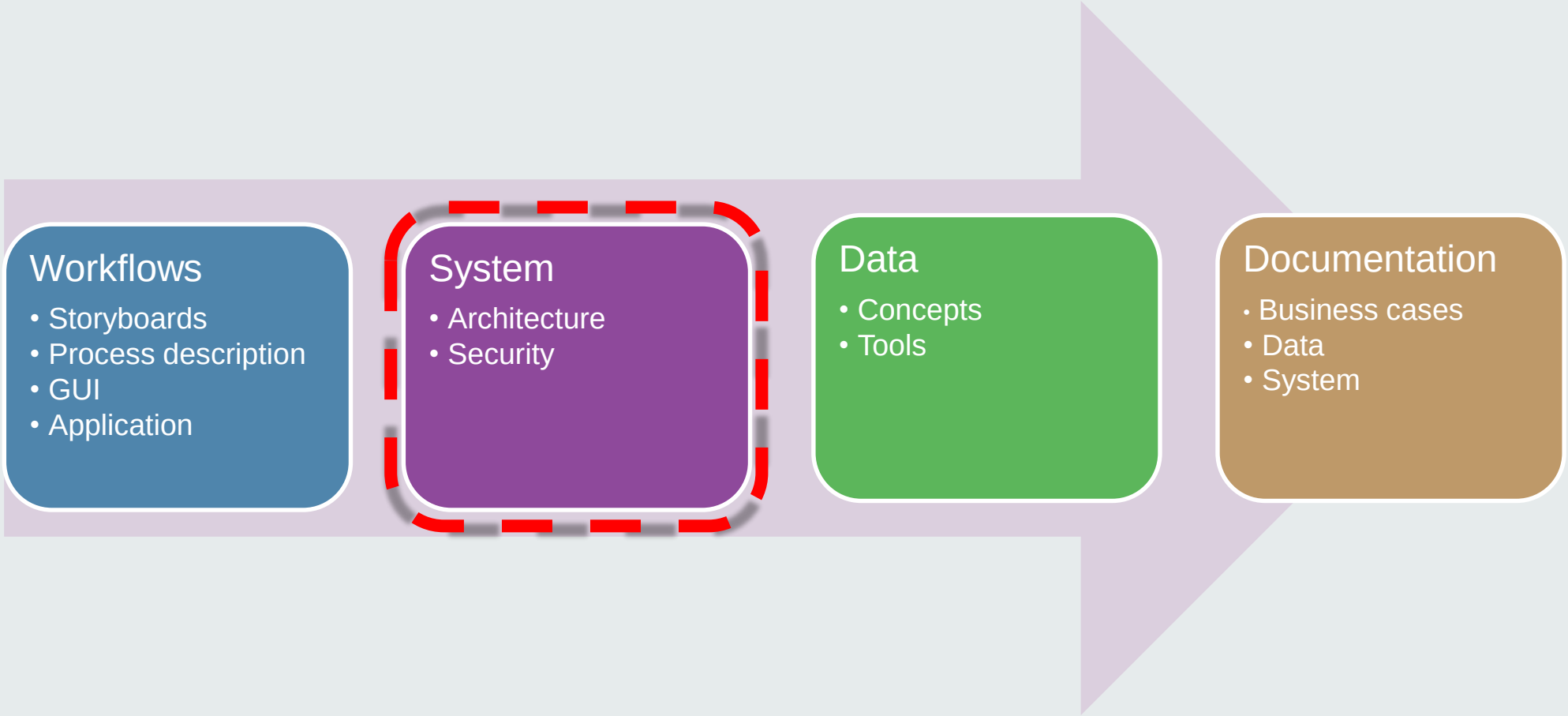


More Functionality

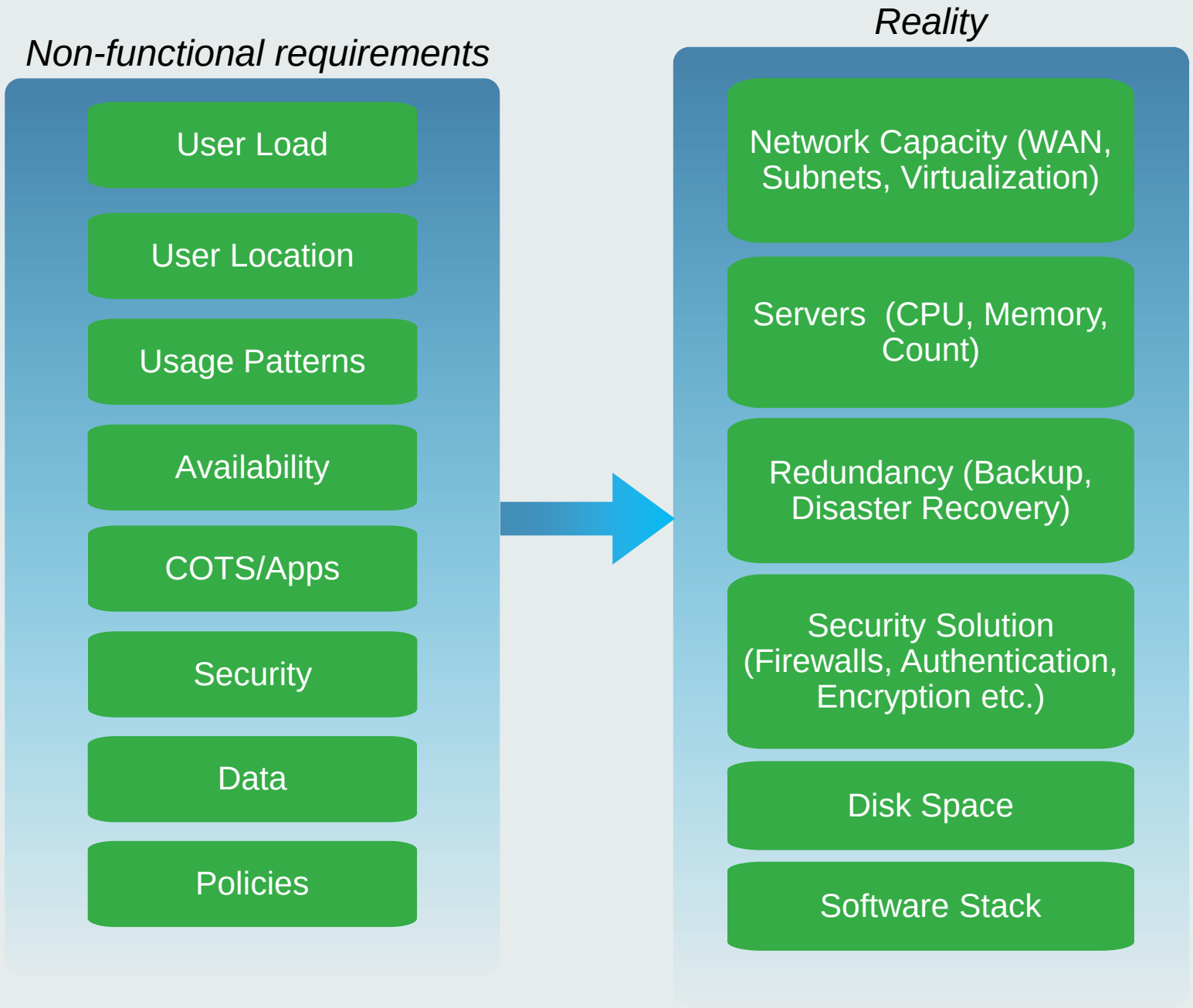
Isn't always better

Unless the users are experts

Design

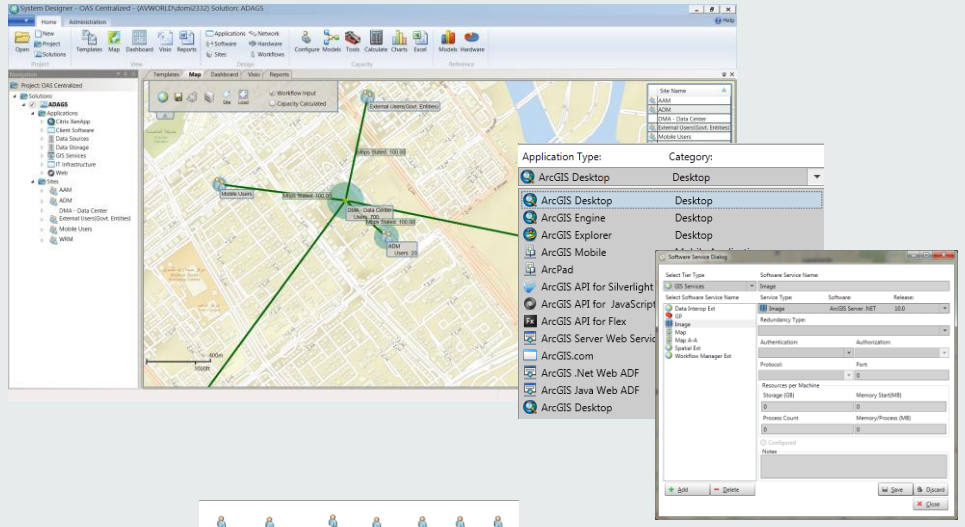

Workflows
• Storyboards
• Process description
• GUI
• Application
System
• Architecture
• Security
Data
• Concepts
• Tools
Documentation
• Business cases
• Data
• System

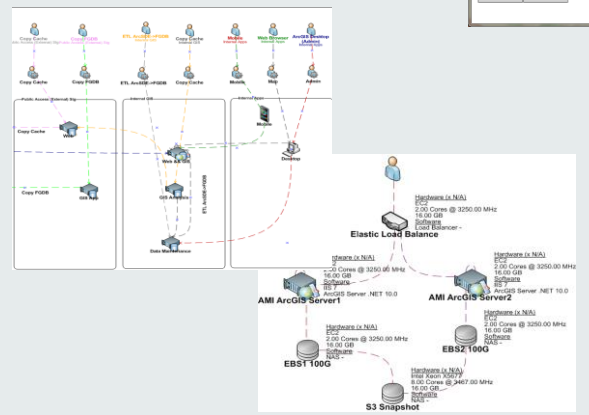
System Architecture: Factors

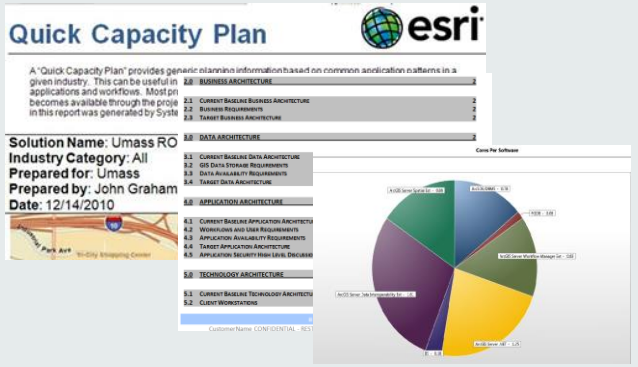


Architecture: Tools & Methodology

- Gathering requirements
- Design tools
- Capacity:
  CPU, Network, Memory
- Reporting on Performance





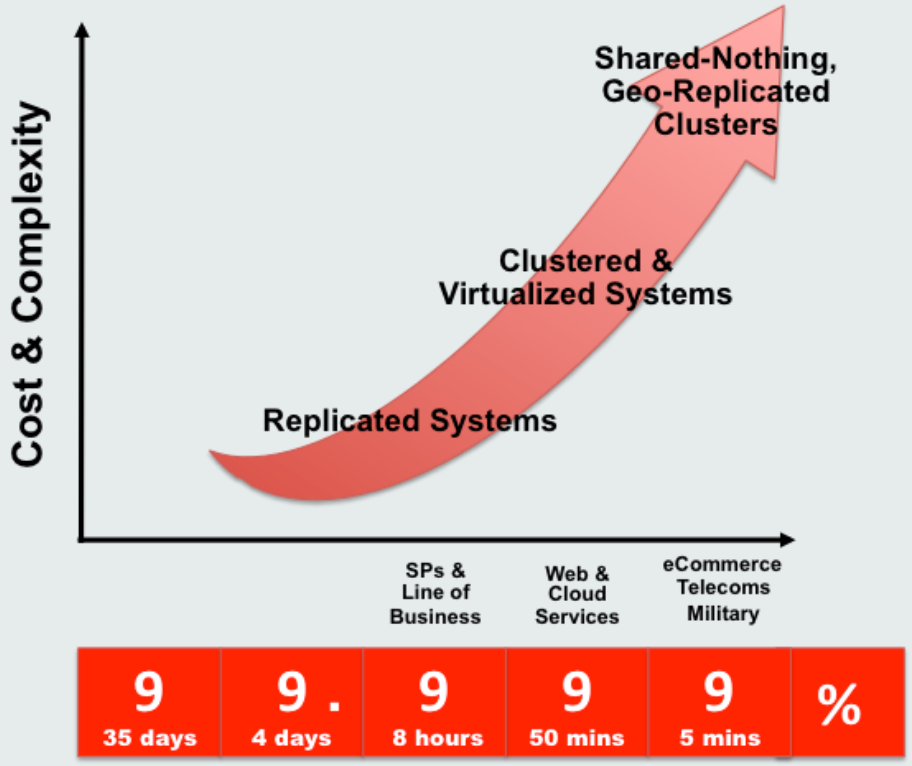


ArcGIS System Designer:
https://www.arcgis.com/home/item.html?id=8ff490eef2794f428bde25b561226bda

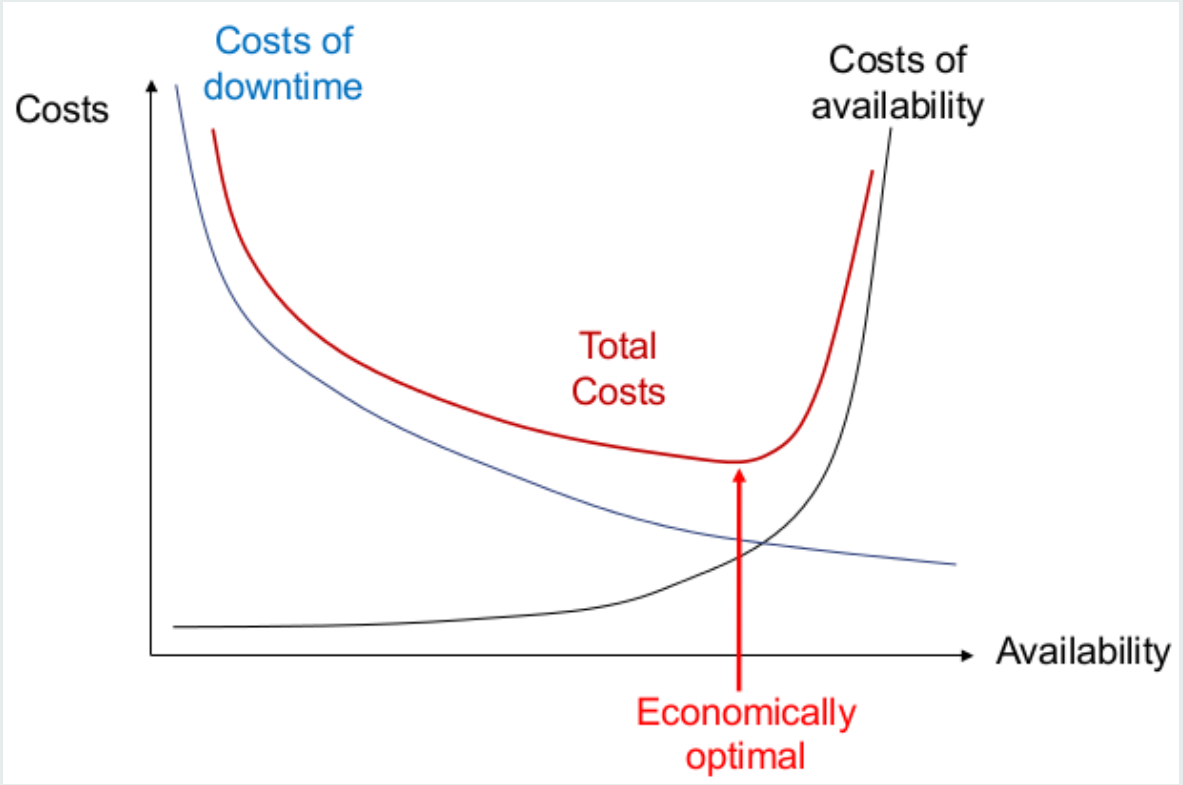
ArcGIS System Monitor available through Esri Professional Services

Architecture: High Availability

- **Engineering for High Availability adds complexity and cost**
  - **Load balancing**
  - **Duplication / redundancy**
  - **Adequate testing, monitoring**
- **How many "9's" do you really need?**
  - **Evaluate costs of downtime against costs of availability**



(MySQL Reference Manual)



System Design: Security

Solutions/Best Practices

Access Control
- Who is allowed to access the GIS
- What is acceptable system usage

→
- Use rules of behavior
- Site banner with acceptable use policy

Audit, Logging
- Monitoring system health and activities
- Minimize performance burden on GIS

→
- Configure tools to strike the right balance

Encryption
- Protecting the confidentiality of your data

→
- Use HTTPS/TLS
- Consider encrypting data at rest

Identity Management
- Uniquely identifying / authenticating users
- Managing user roles

→
- ArcGIS built in-in or your own Provider?
- Single factor or Multi-factor?
- What is your Password Policy?

Intrusion Detection
- How to prevent threats
- How to manage vulnerabilities

→
- Use your Firewall, don't work around it
- Train developers in security practices!
- Implement patch management

http://doc.arcgis.com/en/trust/

System Design: Security

Examples – Identity Management

Implement Password Policy

- At least 6 characters
- Combination of letters, numbers, characters
- Cannot include parts of user name
- Expire after 60 days, cannot be re-used

RANK	PASSWORD	CHANGE FROM 2014
1	123456	Unchanged
2	password	Unchanged
3	12345678	1 ↗
4	qwerty	1 ↗
5	12345	2 ↘
6	123456789	Unchanged
7	football	3 ↗
8	1234	1 ↘
9	1234567	2 ↗
10	baseball	2 ↘

Built-in or 3rd Party Identity Management

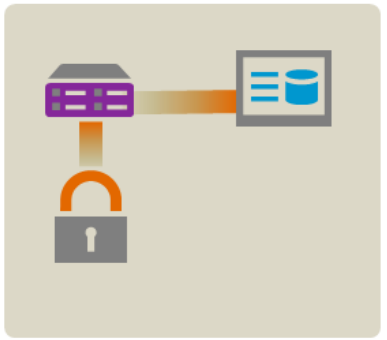
ArcGIS Built-In

GIS Administrator Managed

- Add, configure users
- add roles, grant privileges
- through web interface

…With a small system

3rd Party



IT Managed

- LDAP Server
- AD Server
- SAML 2.0 Compliant Identity Provider

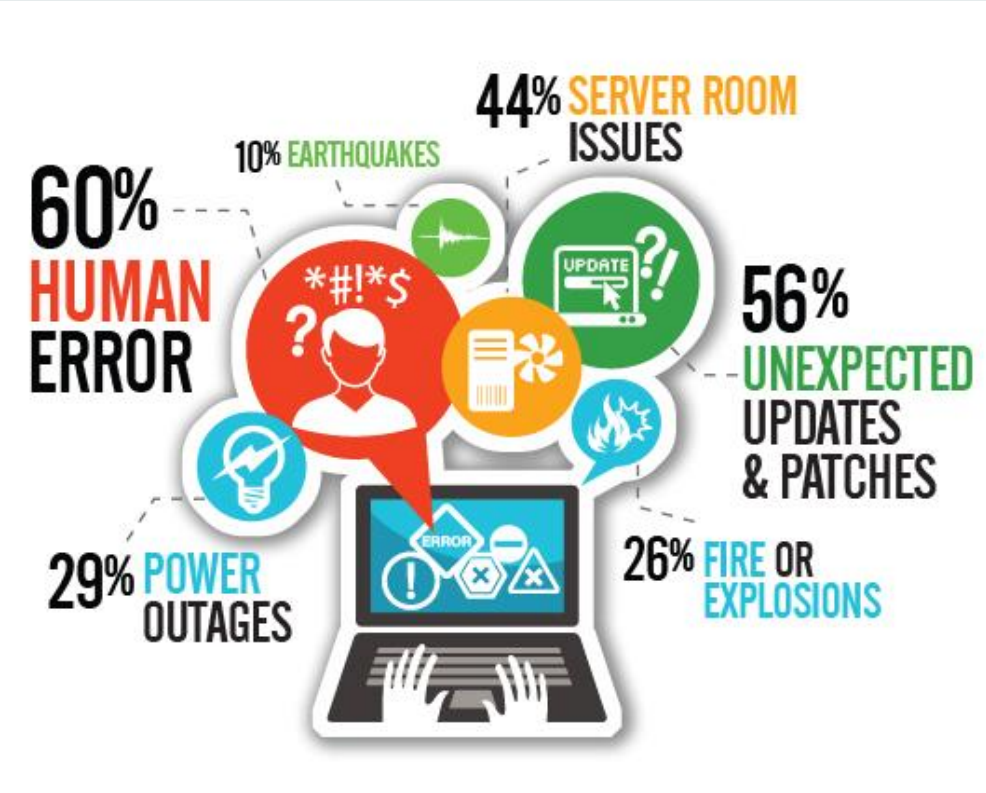
…With a large system

System: Disaster Recovery

- **Is your system mission critical?**
- **In case of a catastrophic failure**
  - **How much time will it take to restore (RTO)**
  - **How much data loss can you afford (RPO)**
- **Develop a plan and test it regularly**
  - **Roles and responsibilities**
  - **Succession**
  - **Backup strategy**
  - **System restoration**

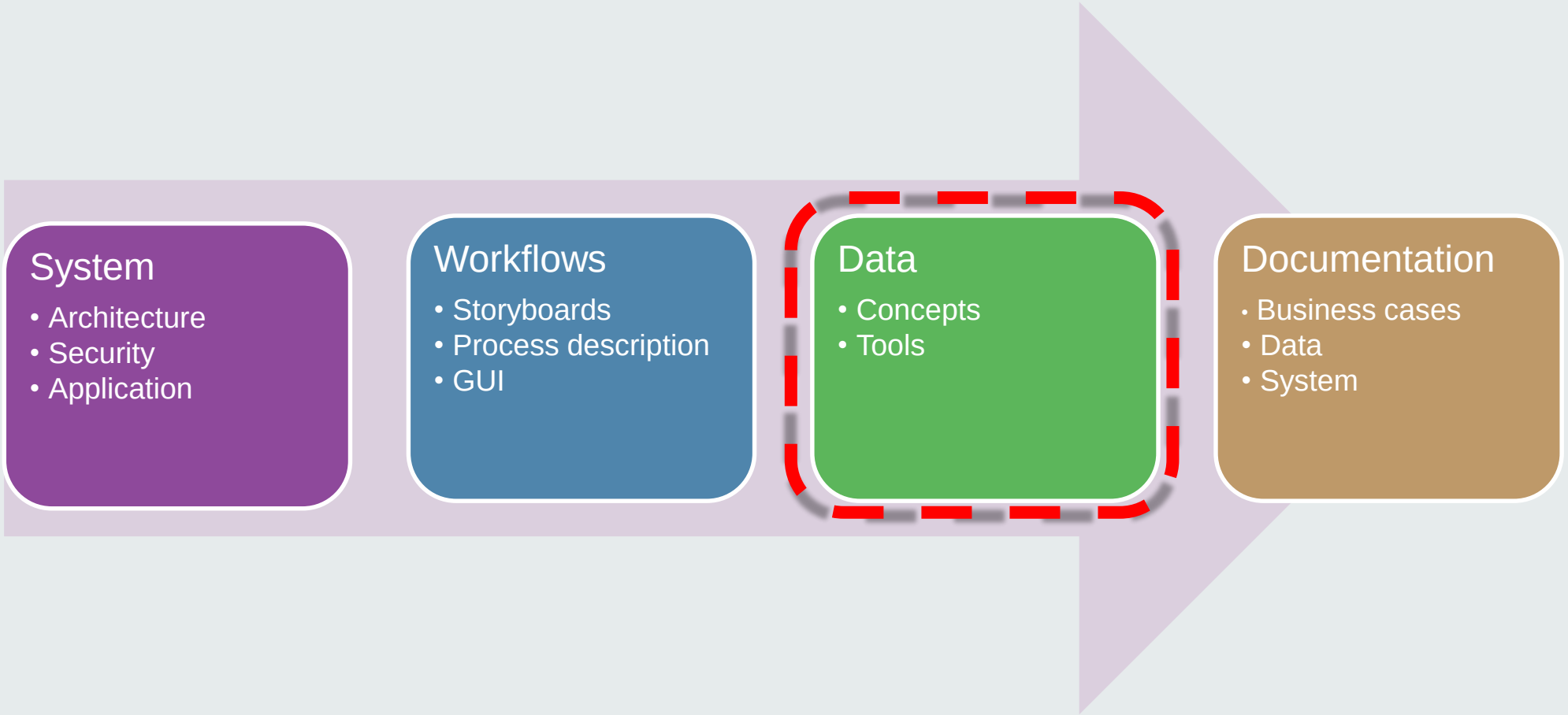
The lower your RPO and RTO, the more redundancy you will have to build into your system!

How many of you have a real RTO/RPO or DR Plan?



Source: Disaster Recovery Preparedness (drbenchmark.org)

Design


System
• Architecture
• Security
• Application
Workflows
• Storyboards
• Process description
• GUI
Data
• Concepts
• Tools
Documentation
• Business cases
• Data
• System

Design the GIS Database

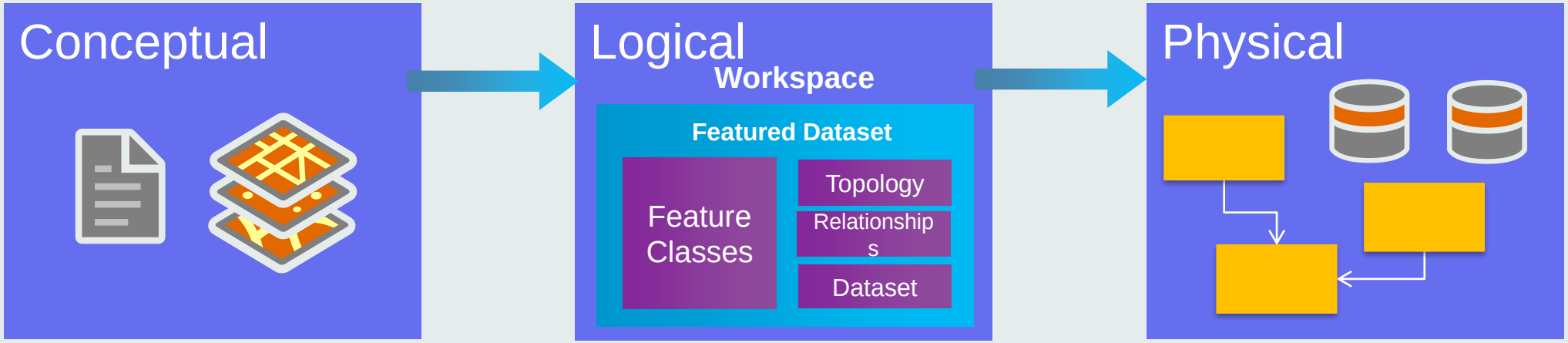
- **Identify data requirements for information products**
- **Collect data inventory**
- **Identify gaps and how to resolve them**
- **Create the data model**

Appendix A
Afternoon
Page 19

Data Requirements Template

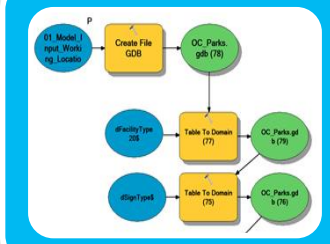
Appendix A
Afternoon
Page 20

Source Data Matrix



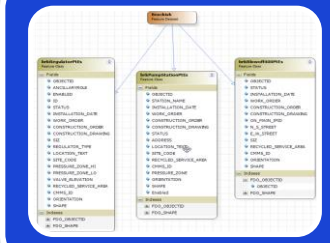
Focus on data you will use and maintain!
Design to the normal not the exception!

Data Design - Tools



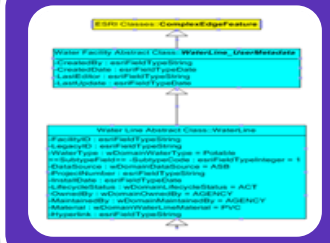
Geoprocessing Models; DDL Scripts

- Repeatable maintenance process
- Initial investment to create the model



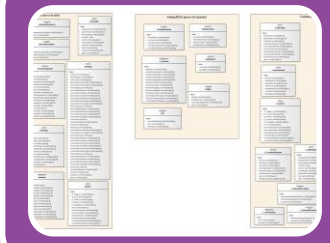
ArcGIS Diagrammer, XRay

- Schema Creation and Documentation Functions
- GDB centric



Visio, UML

- Simple; tool is well known
- Limited support for GDB features,
- No schema creation or reverse engineering (any more)



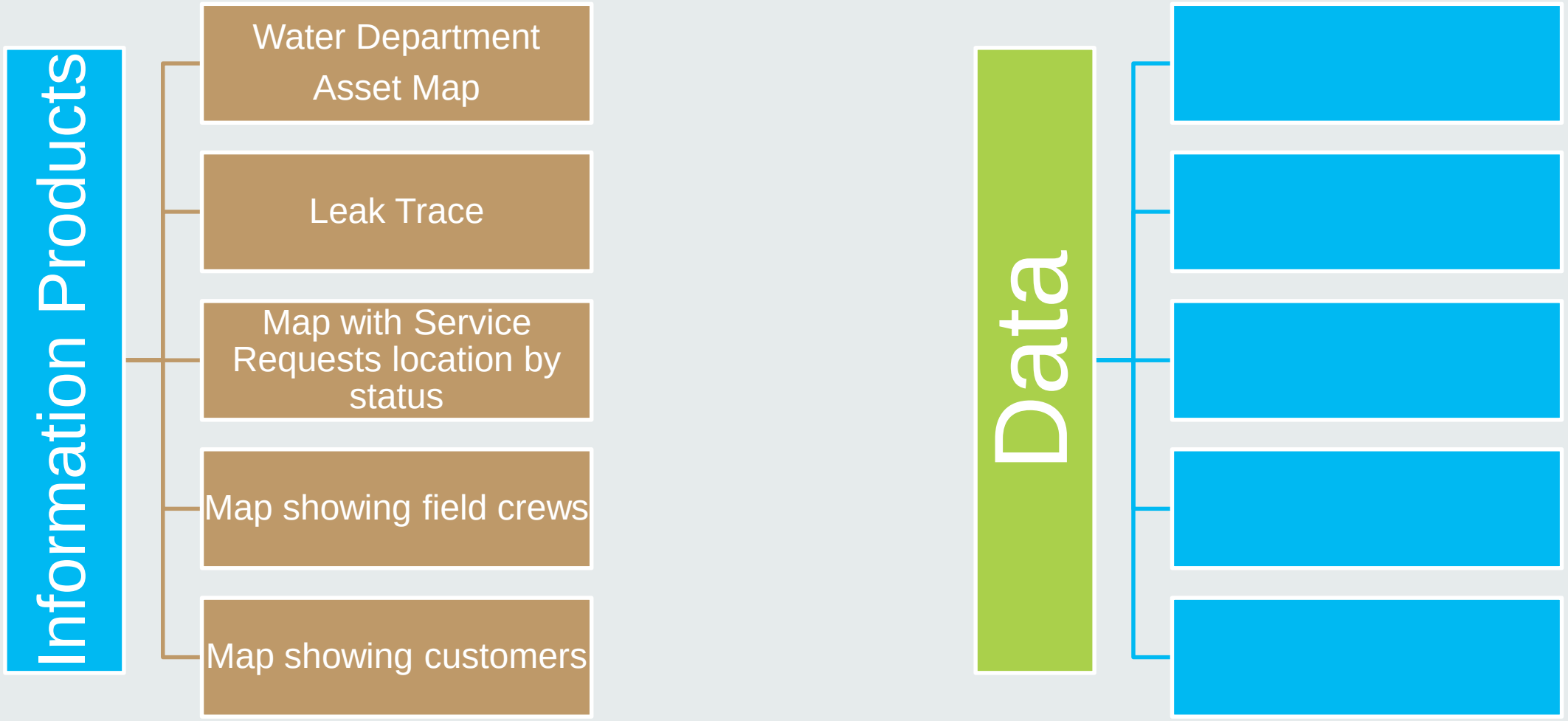
Enterprise Architect, UML

- Full UML Support
- ArcGIS Extension: Import XML Workspace Exports
- Currently better for documentation than design

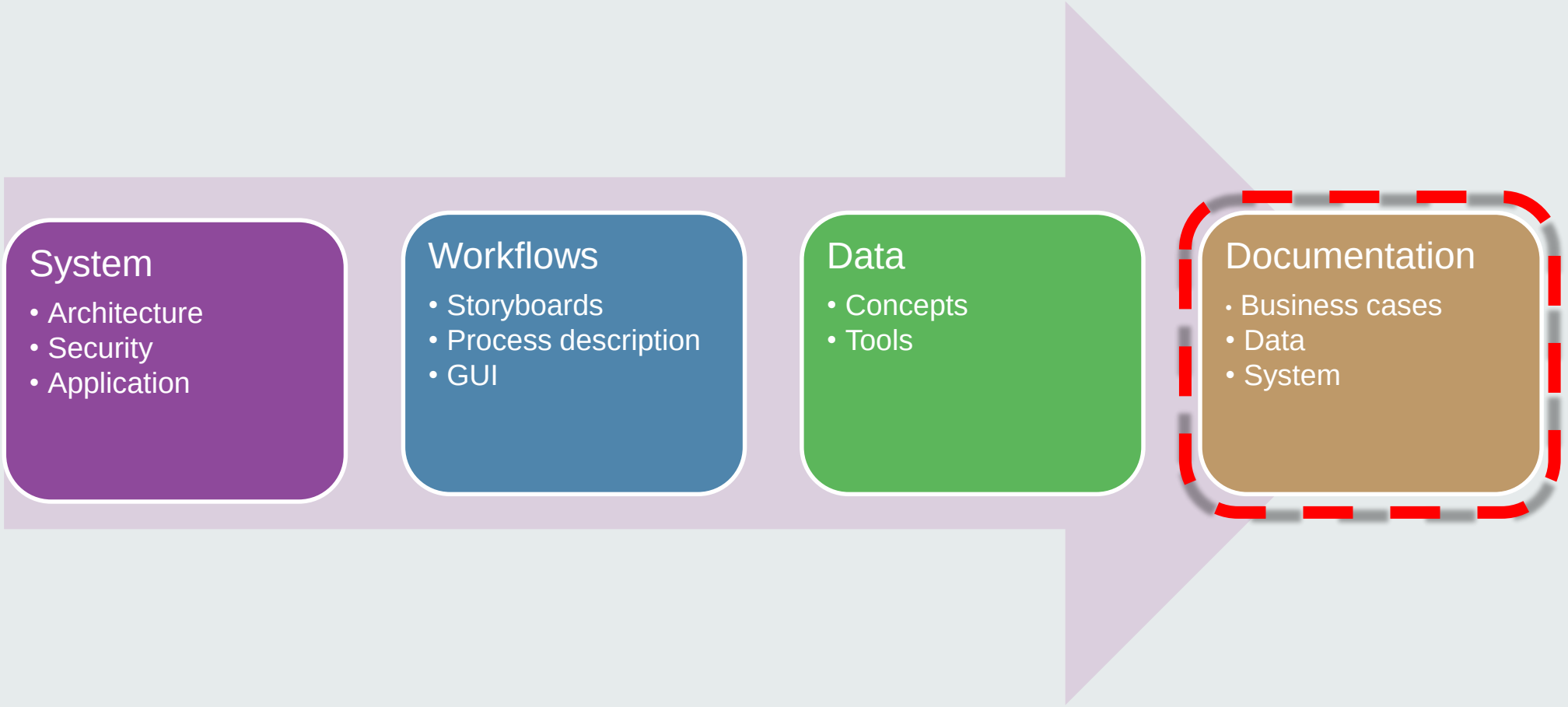
Exercise: List Data Requirements

Exercise– List Data Requirements for Information Products

Example: Water Leak Management



Design


System
• Architecture
• Security
• Application
Workflows
• Storyboards
• Process description
• GUI
Data
• Concepts
• Tools
Documentation
• Business cases
• Data
• System

How much is enough?

Documentation

How Much is Enough? *(for the customer)*

- **Enough to articulate how the business processes will be realized**
  - **Solution architecture**
  - **Concept of Operations (CONOPS)**
  - **User Experience**
  - **User Stories, Scenarios**
  - **Data Layers and Attributes**
  - **Maps, Reports**

Appendix A Afternoon Page 20 — ArcGIS Diagrammer

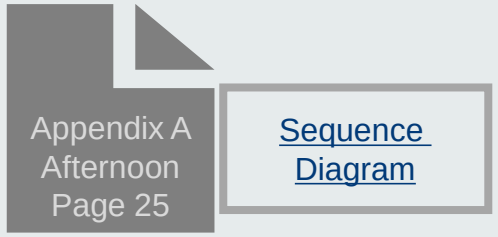
Appendix A Afternoon Page 21 — X-Ray Tool

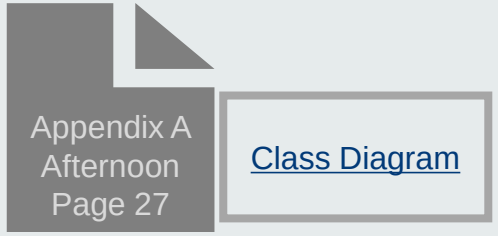
Appendix A Afternoon Page 22 — Use Case Model Activity Diagram

Documentation

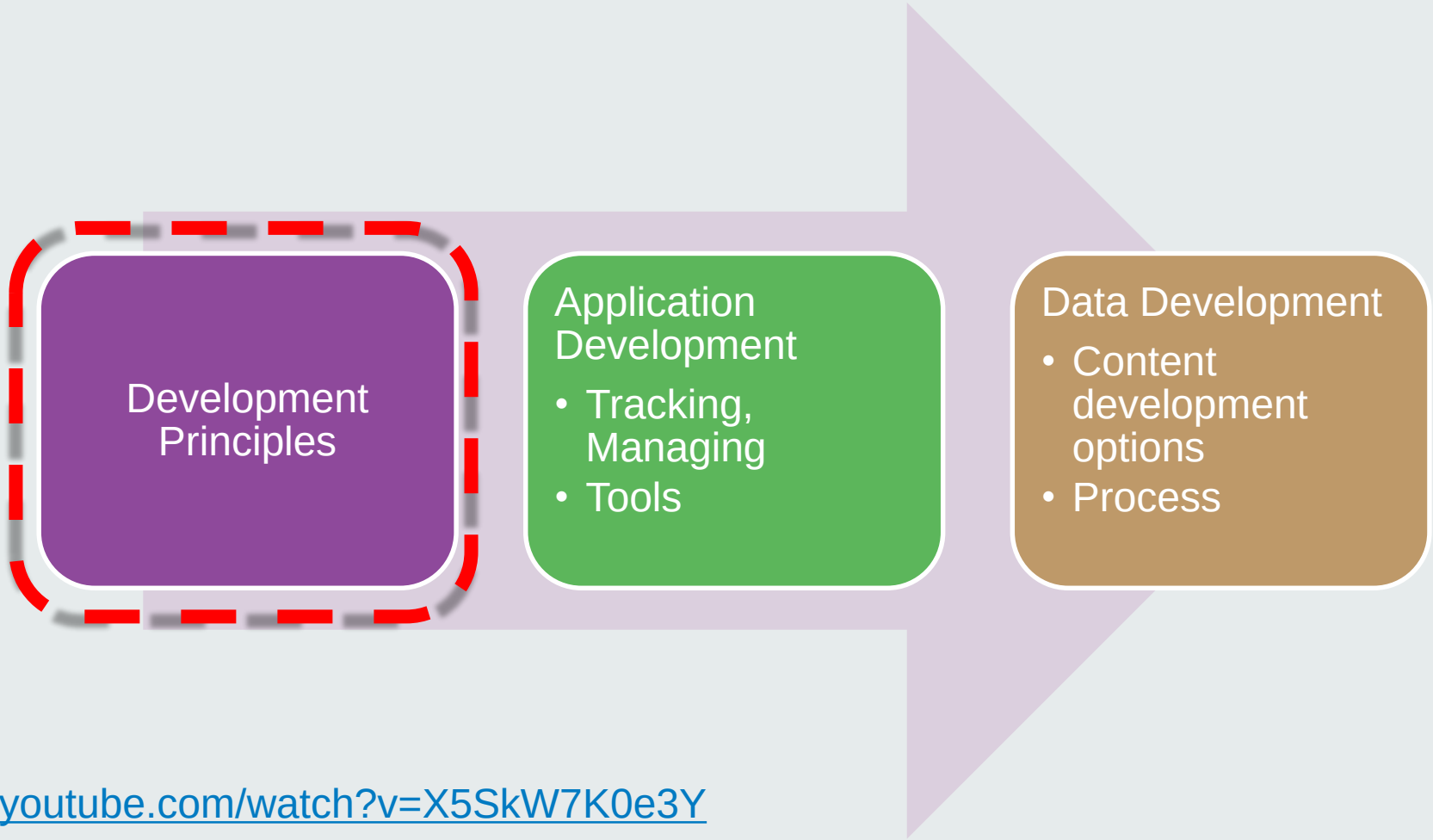
How Much is Enough? *(for technical team)*

- **Enough to determine and articulate (where necessary) how the solution will be developed**
  - **System architecture**
  - **Detailed component breakdown**
  - **Design Specifications for apps and data**
  - **Services**
  - **Prototyping and/or design models for…**
    - ***High risk areas***
    - ***Integration points***
    - ***Key points that are fail points or are non-traditional***
    - ***Prototype → Prove → implement → review cycle***





Configure and Develop your System



https://www.youtube.com/watch?v=X5SkW7K0e3Y

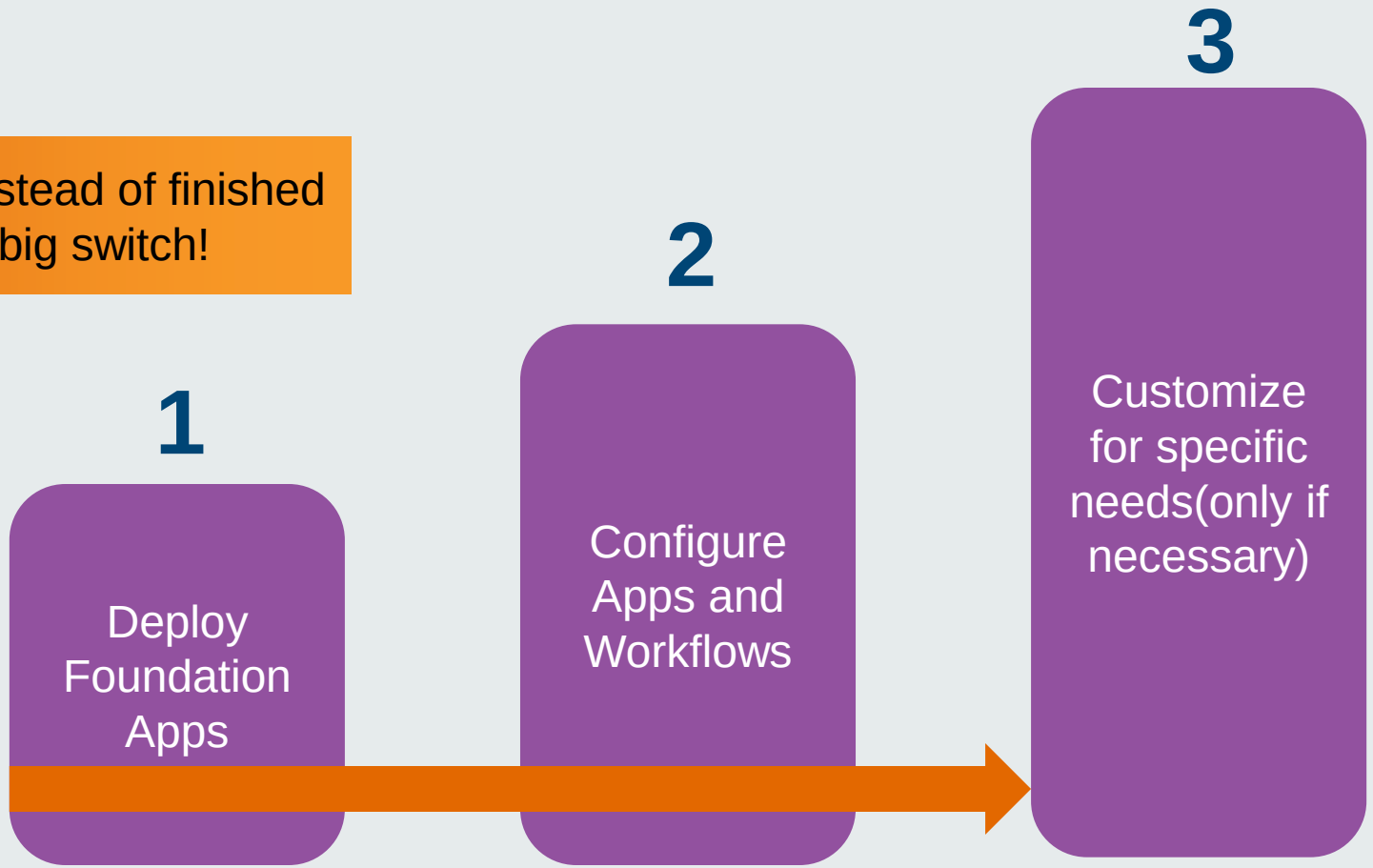
Development Principles

- **Build to your design!**
  - **If change is necessary, inform and explain it to the team, use change management**
- **COTS and configuration first, customization later or when 100% necessary**
- **Track development progress continuously**
  - **Use tools and metrics**
  - **Keep everything visible and review frequently**
- **Use existing and trusted tools for:**
  - **Source control - change control - issue tracking - build automation**

Apps and Information Products | How to Implement Them

- **Use COTS and GIS-ready data to deploy Initial Operating Capability (IOC)**

Presenting IOCs instead of finished products is a big switch!



Development Principles

Tools are your Friends

- Work Item Management
- Source Code Repository
- Build System
- IDEs

Use Coding Standards

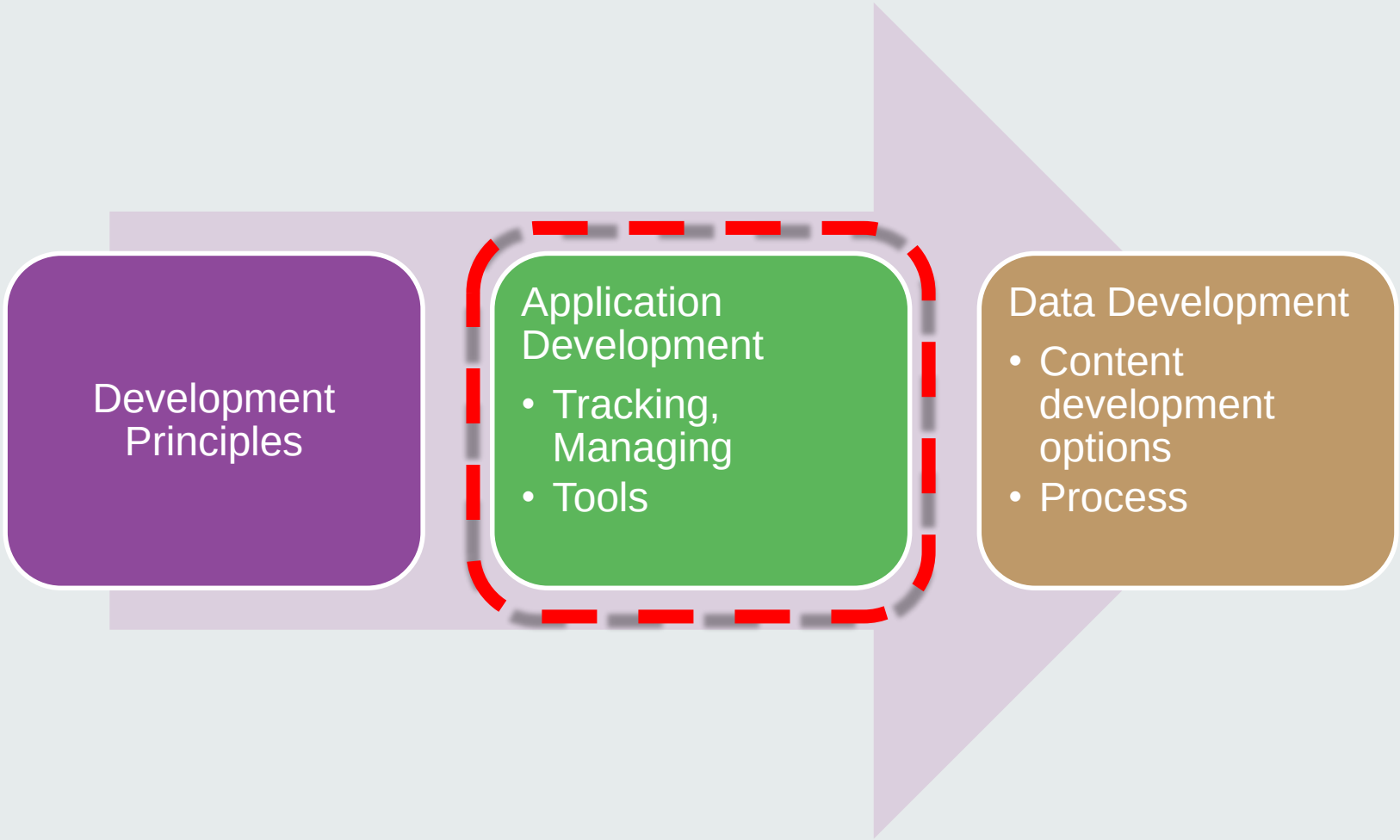
- Logging
- Exception handling
- Naming conventions
- Code Reviews

Enforce Unit Testing

- By developers
- Use automation
- Test design
- Test robustness of code

http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments

Configure and Develop your System



Application Development - Tools

Source Control
- TFS
- git/GitHub
- Subversion
- Mercurial

Build
- TFS,
- JIRA
- Travis
- Ant
- Maven
- Grunt
- Less

Work Item Management, Issue / Change Tracking
- TFS,
- GitHub
- JIRA
- Rally, Trello
- Bugzilla
- OnTime
- Remedy
- Lighthouse

Development Environment

IDEs
- Sublime Text
- Visual Studio
- Eclipse
- Technology Stack dependent

Consider a complete lifecycle tool (integrates source control, build, issue tracking)

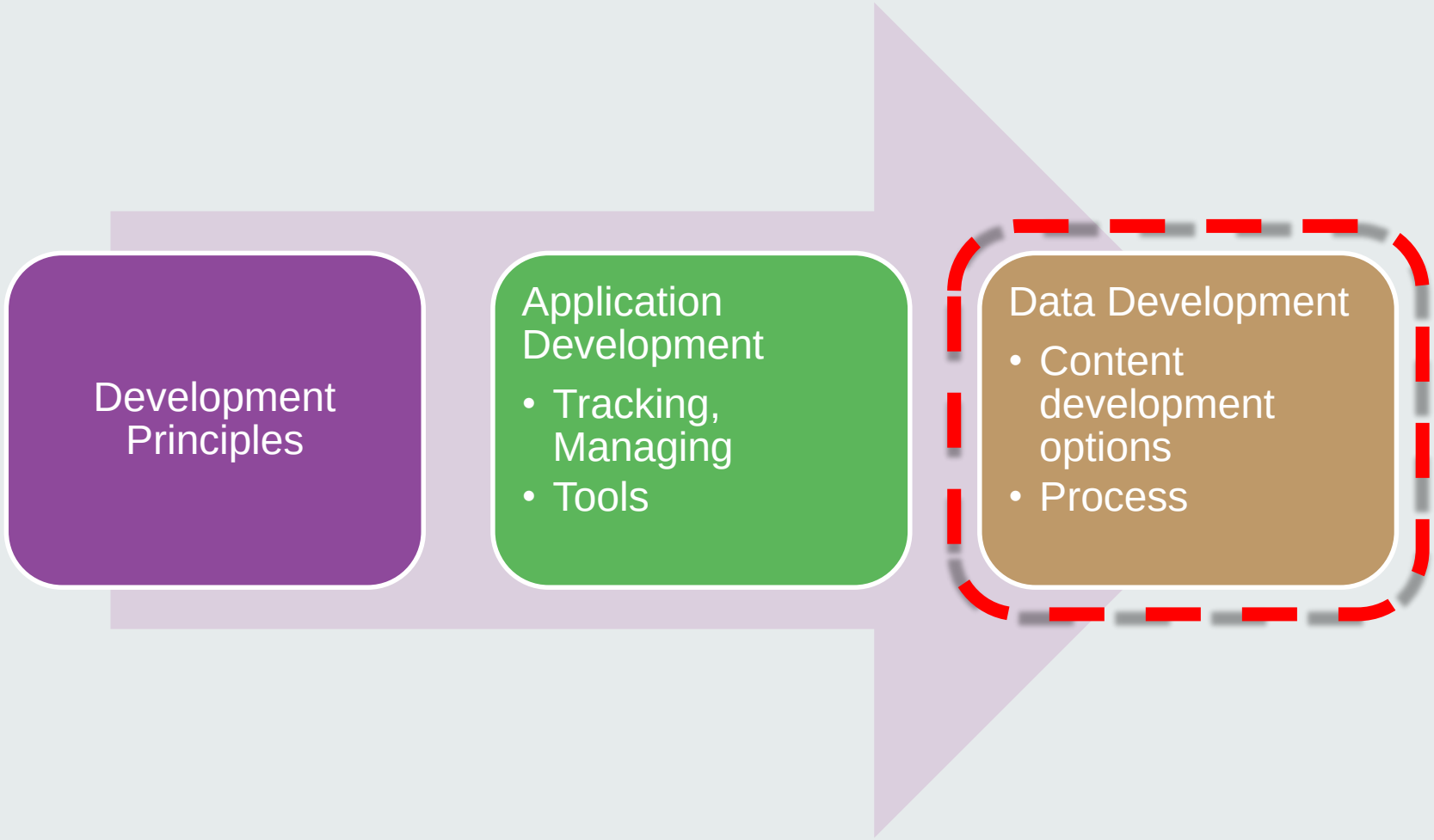
Application Development - Server-Side Technology Stacks

- **Options:**
  - **Microsoft .NET**
  - **Java**
  - **Node.js**
  - **Also:**
    - **Ruby on Rails, Django, …**

Application Development - Recommendations

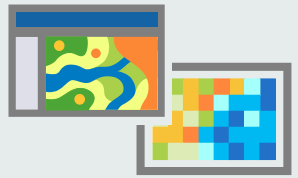
- **JavaScript is the best bet for native-looking apps**
- **Python is not only for geoprocessing but for web services, hosting web apps with Flask and other cases**
- **Mix and match languages to suit the need and suit your team's experience**
- **With web services, you can use any language / any technology stack to access the services once they exist.**
- **Leverage existing APIs or wrapper libraries**
  - **ArcREST**
  - **arcgis-node**
  - **node-geoservices**

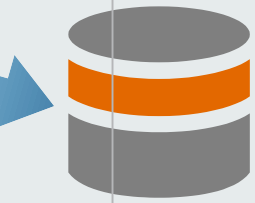
Configure and Develop Your System



Data Development

What are the Options for Developing Data Content?



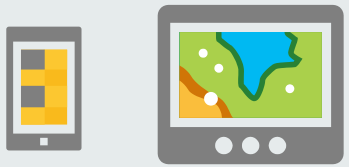


Migration – Move data between GIS Platforms

- AutoCAD, Microstation
- to ArcGIS
- Shapefile to GDB

Conversion – Automate new geospatial data

- Hardcopy Maps to GDB
- Imagery to vectors



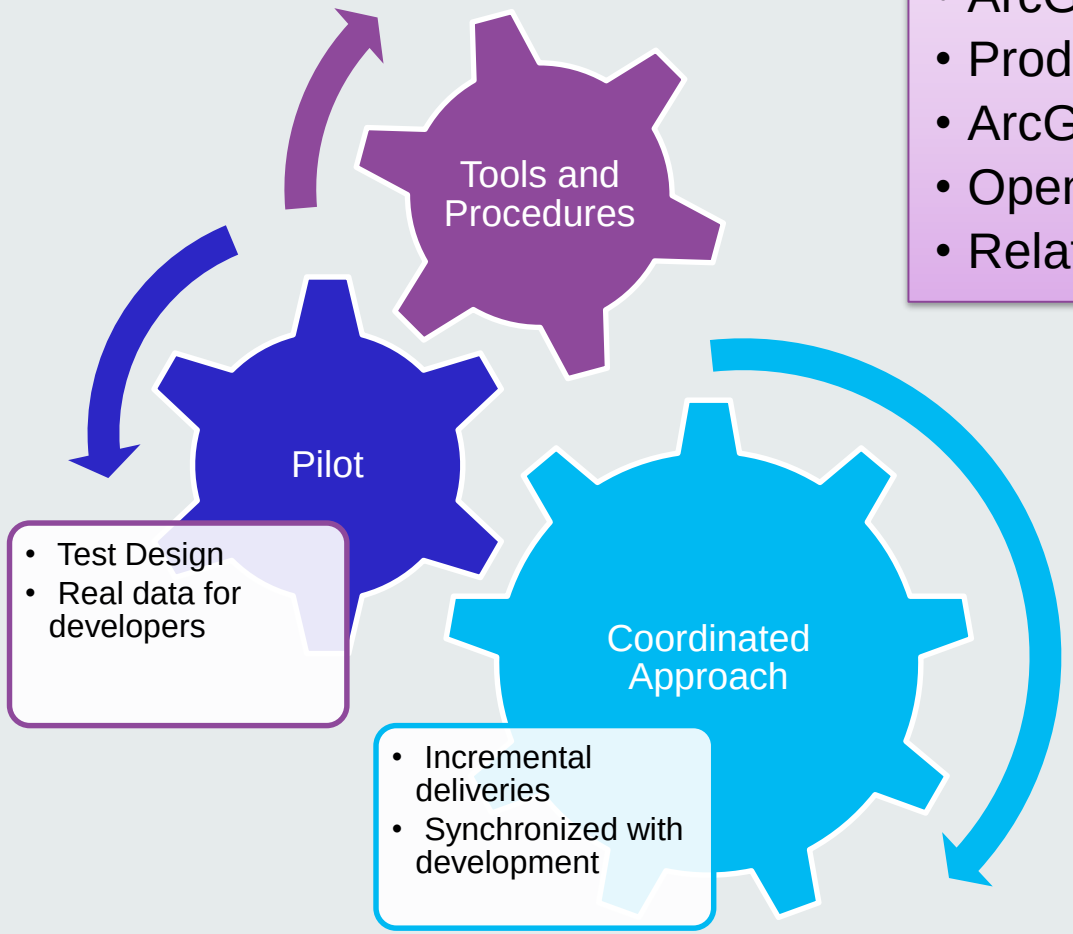
Collection – New data from remote platforms

- Imagery
- GPS
- Surveys

Appendix A
Afternoon
Page 29

Data Migration Procedures Document

Data Development



Tools

- ArcCatalog
- ArcGIS Data Interoperability Extension
- Production Mapping Data Loader
- ArcGIS Data Reviewer
- Open Source data production
- Relational Database Geometry storage

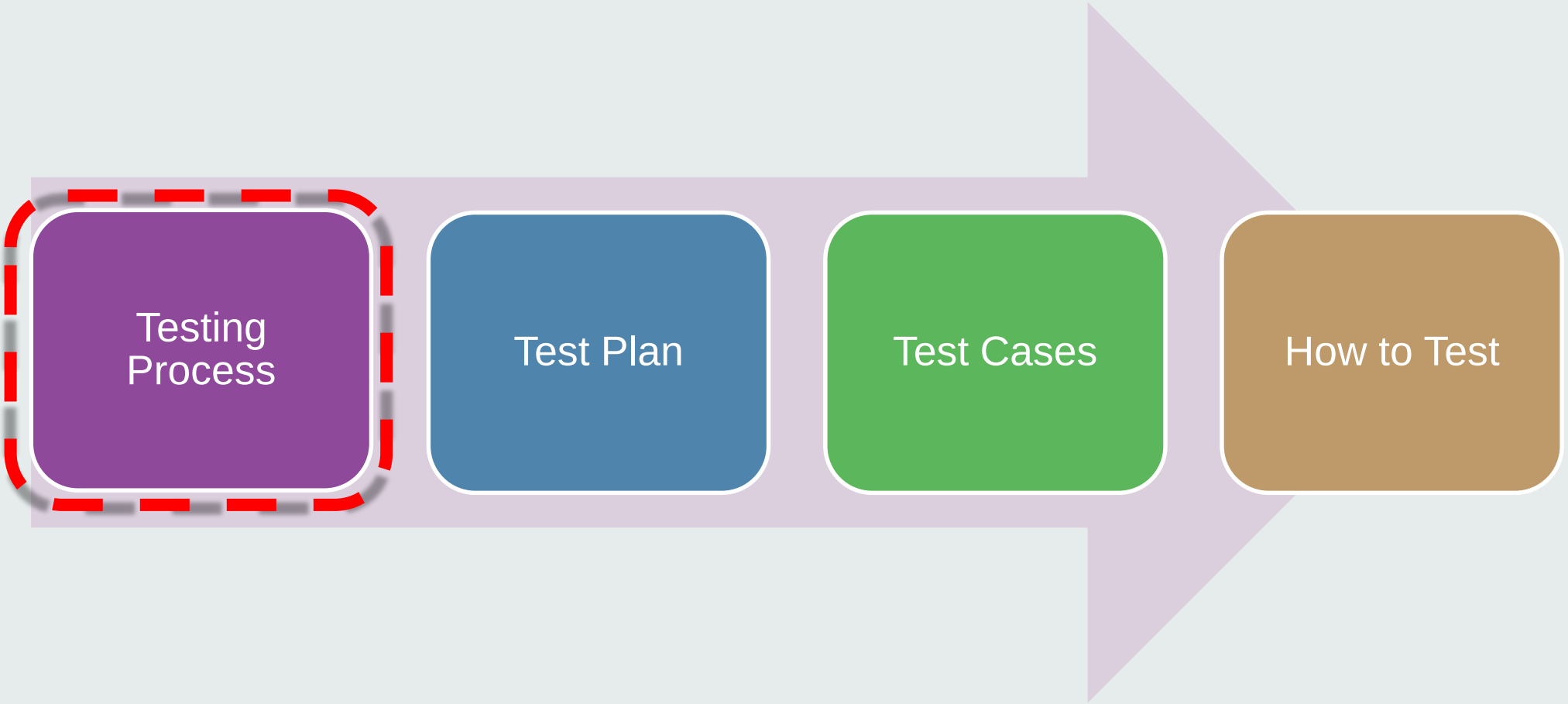
For best results:

Let your data evolve with your Apps and information products!

Development - Review

- **Formal process is key**
- **Strive for transparency in the process**
- **Engage the leadership team and sponsors**
- **Coordinate database and application development**
- **Keep developers developing and testers testing**
- **Stay engaged with customer or project owner**
- **Test early, release as often as is possible and prudent.**
- **Code Reviews**

Test your system



Testing Themes

- Test the right things at the right time
- Focus on your deliverables
- Develop good test cases
  - Prove requirements are met
  - Fit the importance of the requirement
  - Reflect workflows
- Use an (approved) test plan
- Tests need to be realistic and fair

https://www.youtube.com/watch?v=mafRRWKSwu0

Testing Process

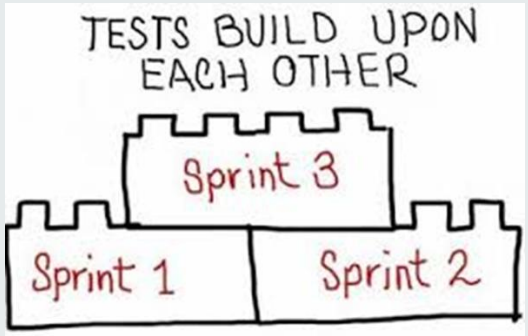
Testing Works Best as an Integrated Process!

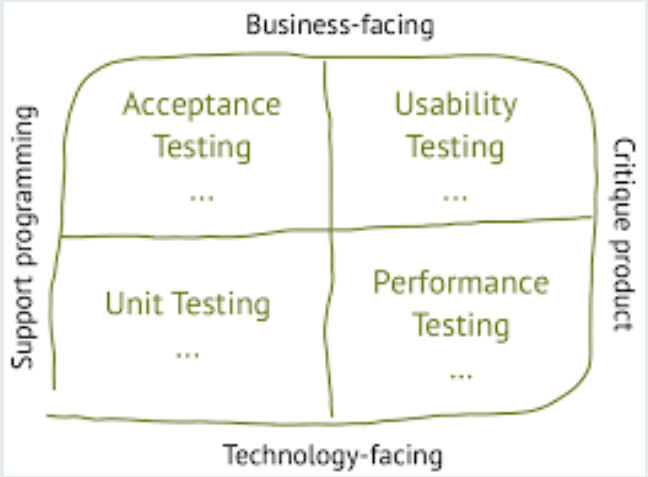
Testers, Designers, Developers
Collaborate during test case development (design stage)

Developers
Develop and unit test single components

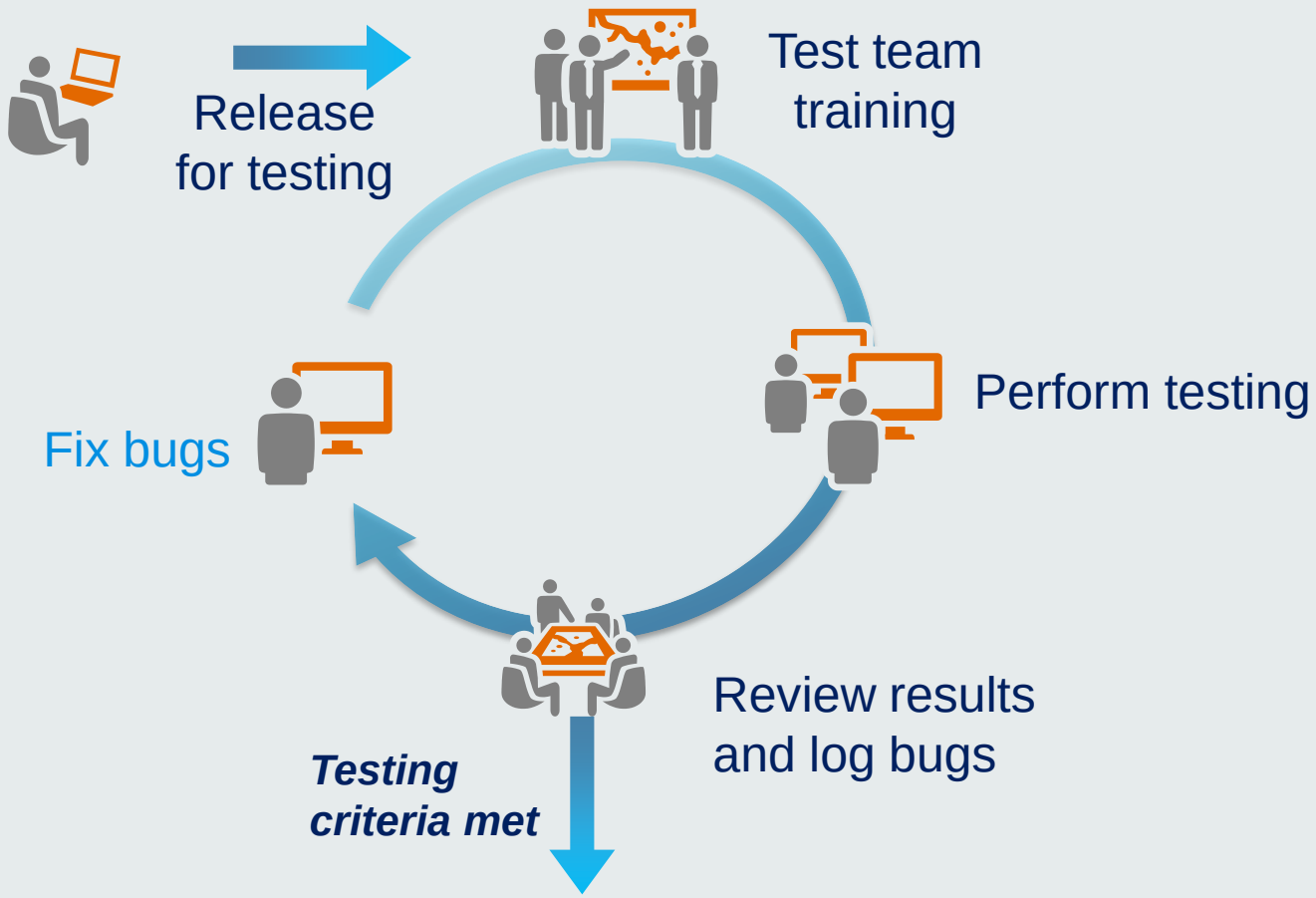
Testers
Stakeholders, End Users
Test in a separate test environment

Test Plan and Test Cases





Testing Process

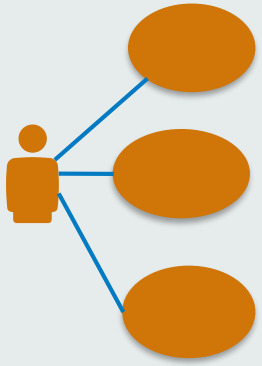


Sample Design Artifacts for a Leak Management Isolation Trace Tool

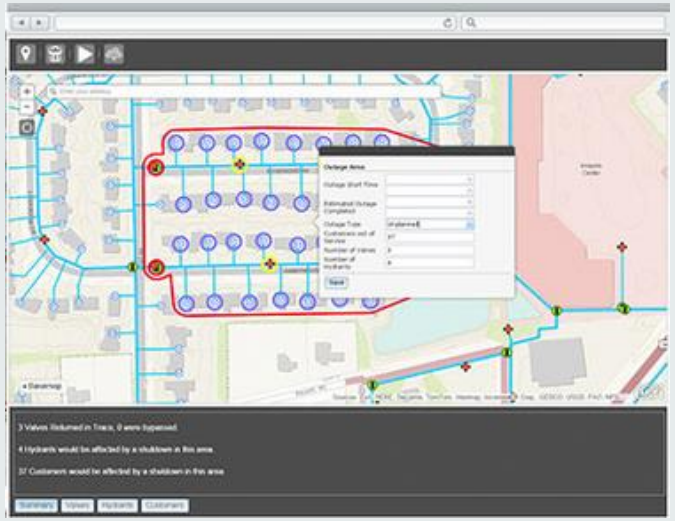
User Stories

As a Field Crew Member
I want to run an leak isolation trace
So I can identify affected service area, customers, and assets

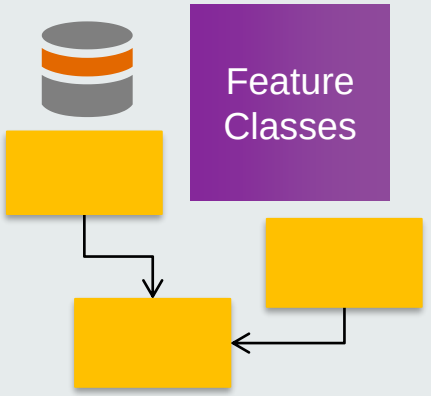
Acceptance Criteria



User Interface Prototype & Information Products Design



GIS Data Layers



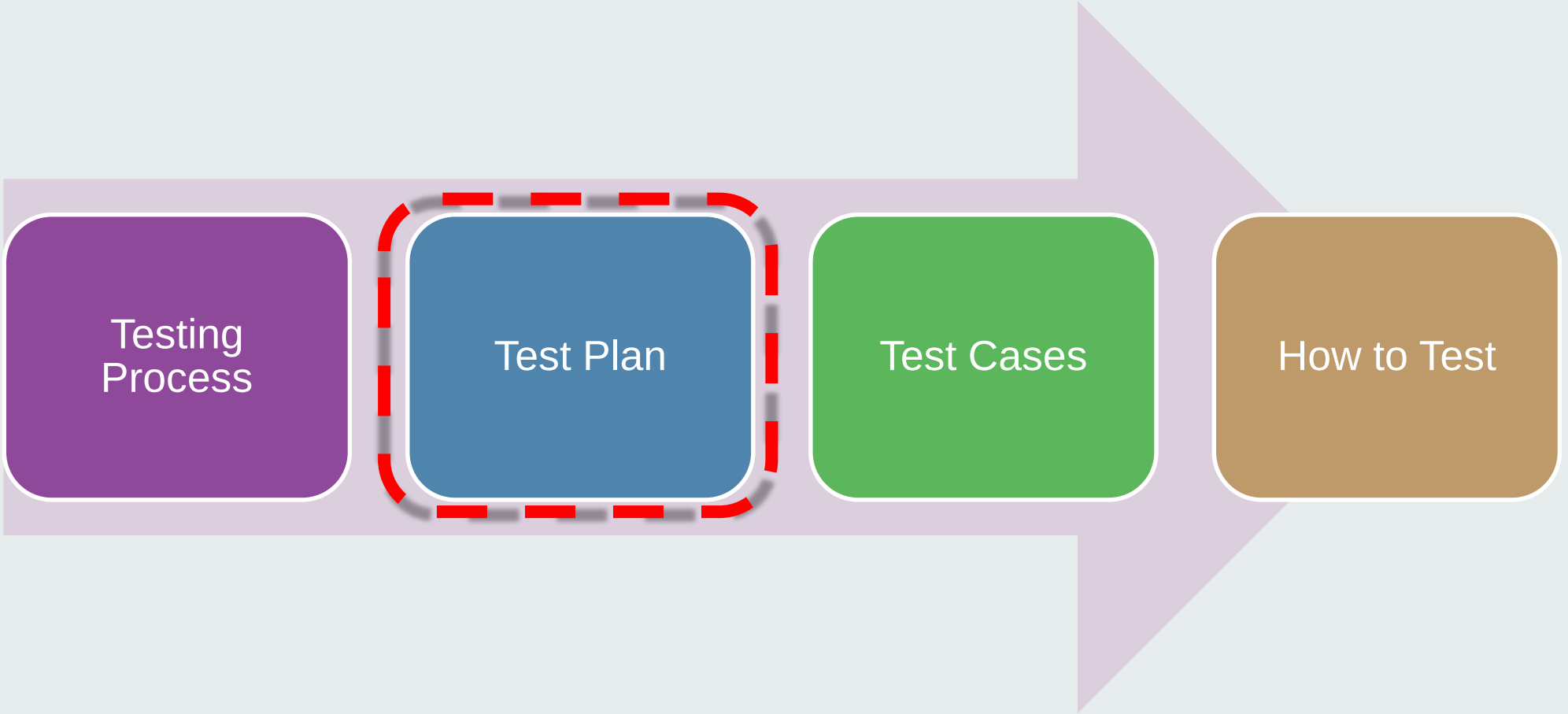
Develop Design Artifacts Specific to the Sprint

Configure Trace Tool
Execute Trace
Save Trace

Summary Report
Valve Report
Hydrant Report
Customer Report
Trace Map

Pipes
Valves
Hydrants
Service Locations
Work Order Locations
Basemap Layers

Test your system



Test Plan

Typical Test Plan Outline:

- Your testing strategy including:
  - Overview of the solution
  - Objective with the testing
  - Scope of the testing
  - Sequence of testing
- Acceptance procedure and criteria
- Test environment to be used
- Test data and test area to be used

Appendix A Afternoon - Pages 32 - 33

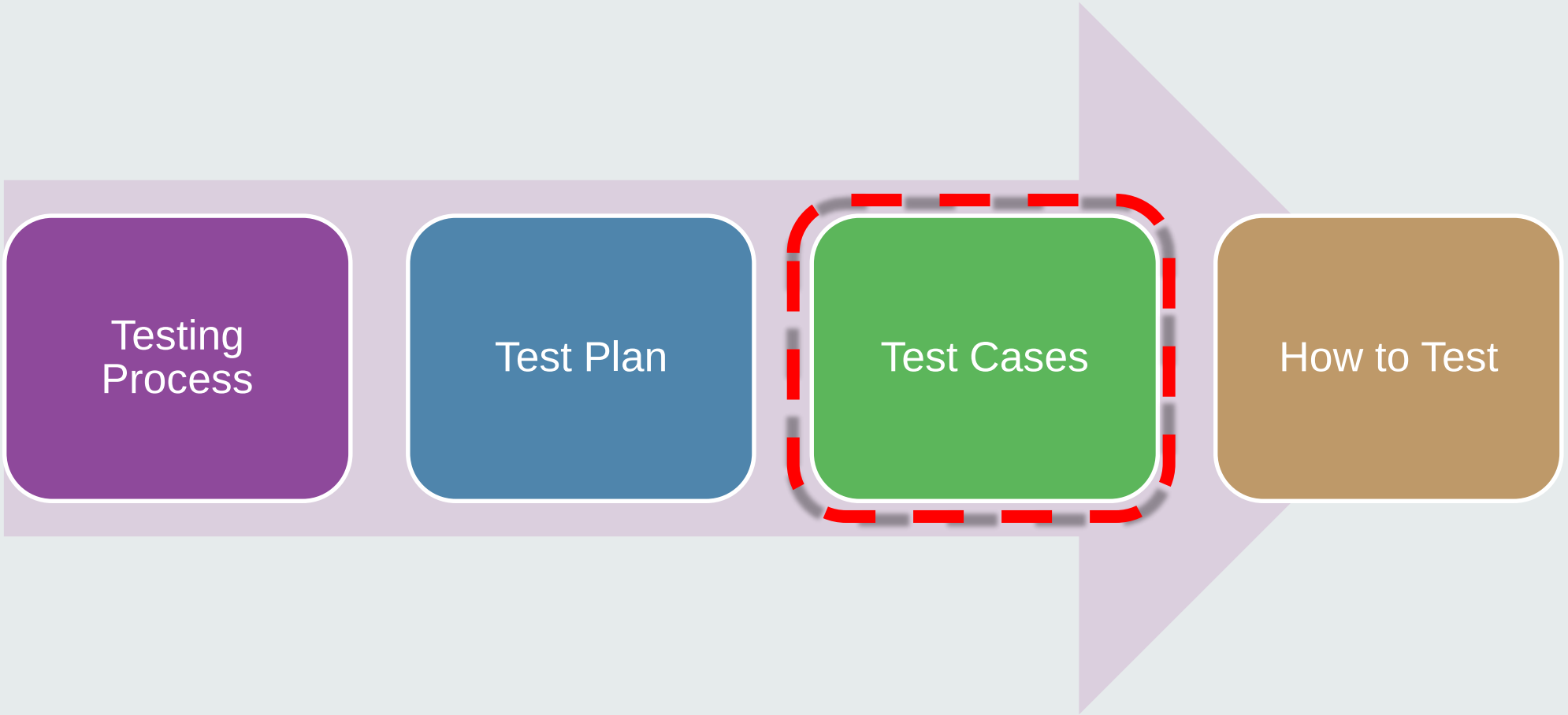
Test Plan Templates

Appendix A Afternoon Page 35

Acceptance Criteria

The test plan is your 'battle plan' for acceptance

Test your system



Develop Test Cases

- Test cases are procedures for verifying requirements
- Address all requirements
  - Functional
  - Non-Functional
- Good Test Cases:
  - State the purpose of the test with reference to requirement
  - State pre-conditions
  - Clearly and objectively outline the steps to verify/validate a specific outcome
  - Unambiguously list the expected result or outcome of the test determining a pass/fail of the test

Appendix A
Afternoon
Pages 37
to 39

Test Case Template
Test Script

User Stories and Testing

…Think back to the invest model…Think Traceability

- **I** Independent
- **N** Negotiable
- **V** Valid
- **E** Estimable
- **S** Small
- **T** Testable

User Story Template

As a [User Role}
I want to [Desired Feature]
So that [Value/benefit]

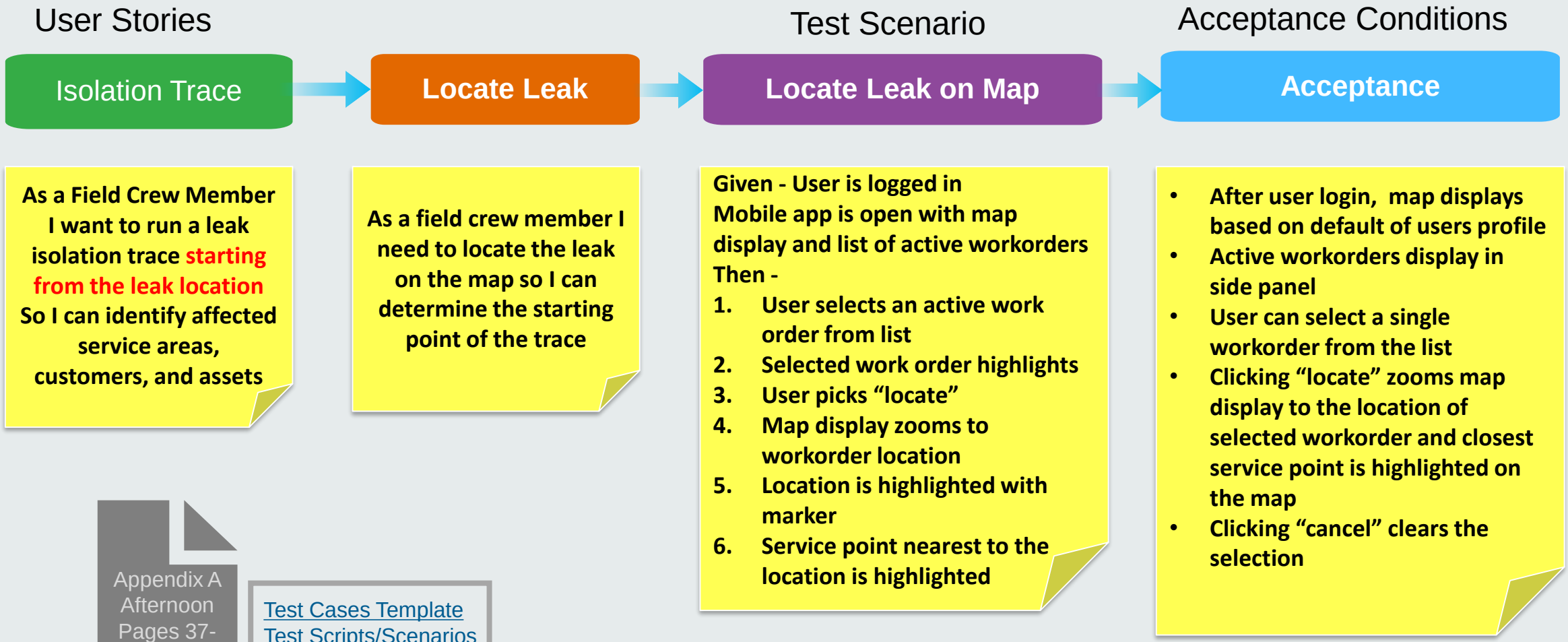
Focus on
- **Value to the user**

Testing Template

Given [Some Context]
[When] some action is carried out
[Then] a particular set of observable results should be obtained

Focus on
- **Pre-conditions, Post-conditions**
- **Step-wise descriptions**
- **Testable outcomes**

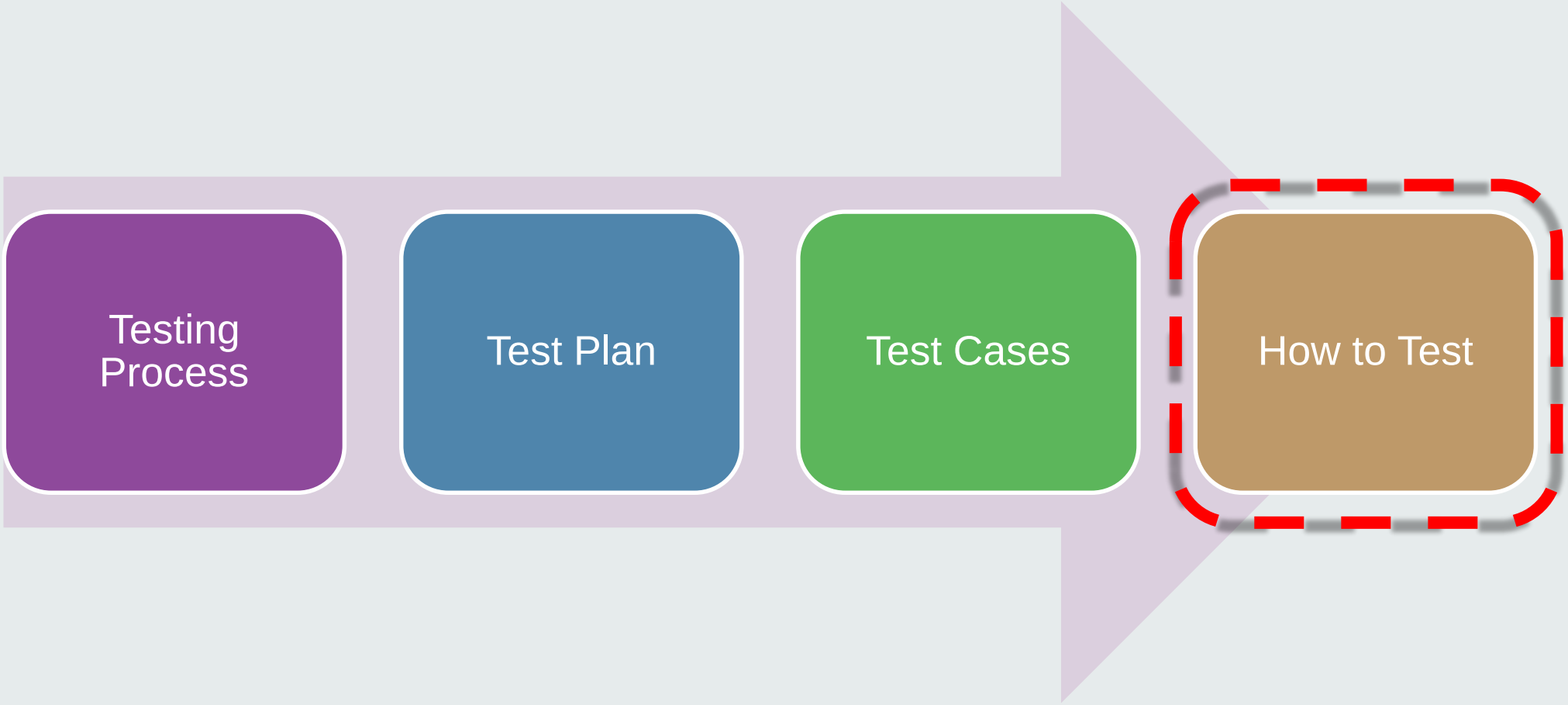
Example: Test Scenario for a Water Leak Isolation Trace User Story



Appendix A
Afternoon
Pages 37-39

Test Cases Template
Test Scripts/Scenarios

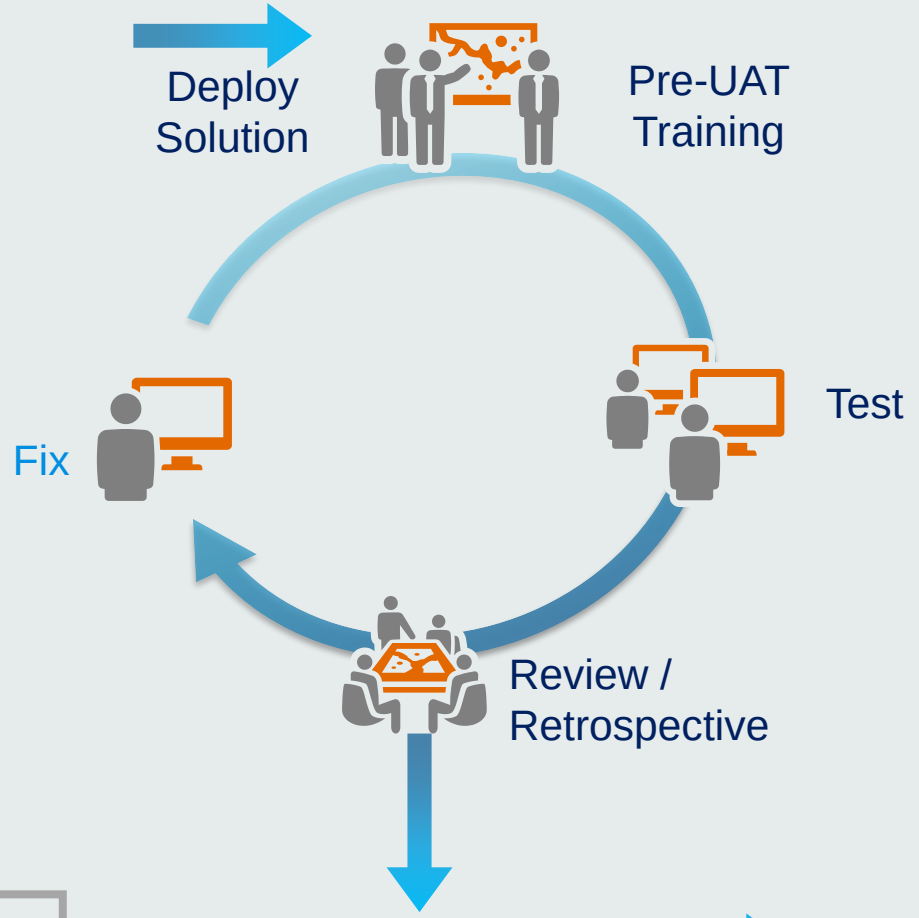
Test your System



How to Test

Acceptance Testing Process

- **Your site or client site?**
- **How much time?**
- **When to train?**
- **What happens after the test?**
- **Is on-site support necessary?**
- **How to track and manage defects?**

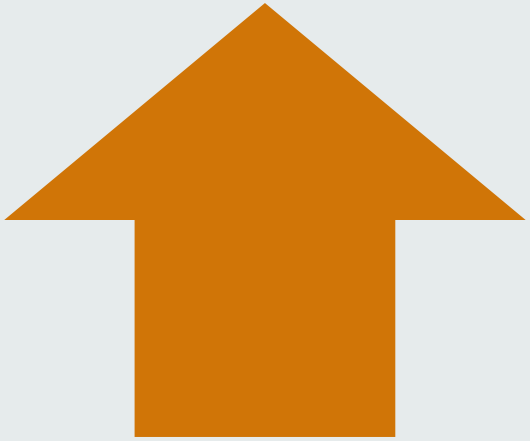


Appendix A Afternoon Page 44 — Defect Tracking

Appendix A Afternoon Page 45 — Test Results Report

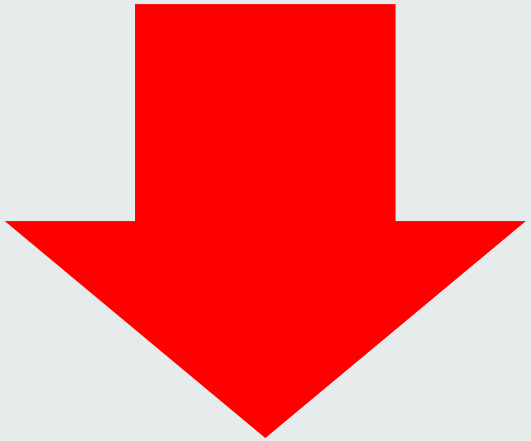
Testing process

How Do You Evaluate Software Quality?



Measure and analyze against acceptance criteria

- Defect metrics
- Status meetings



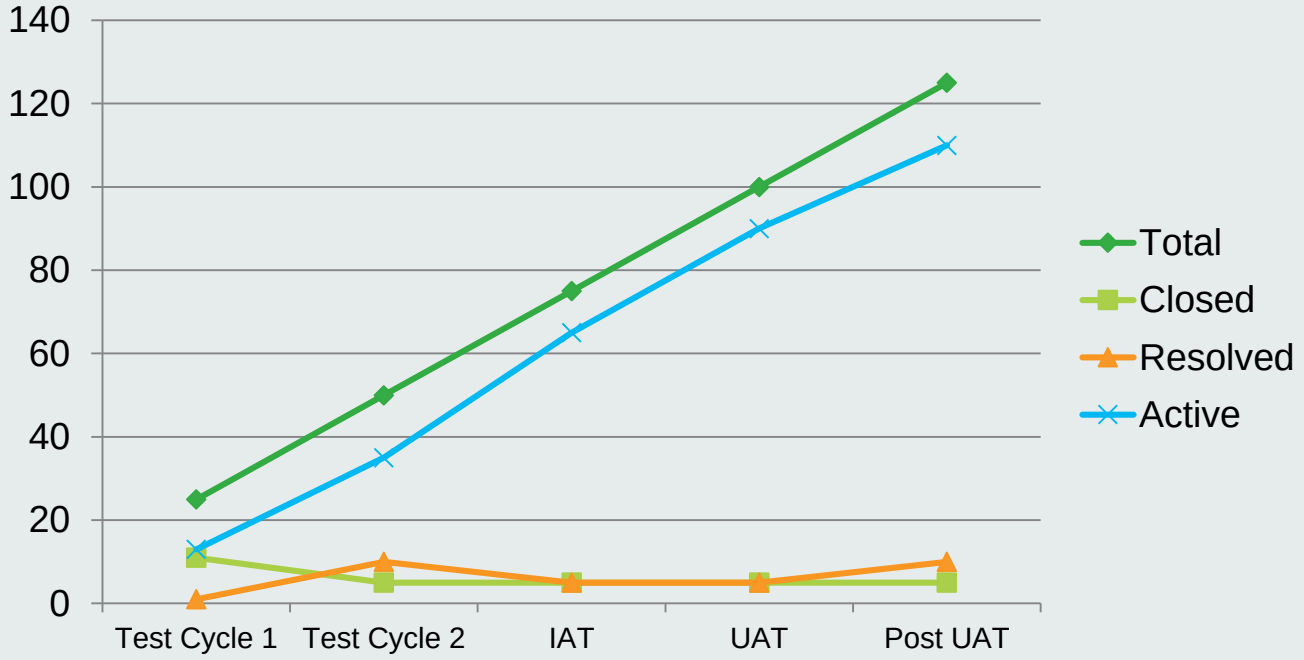
Anecdotal evidence from developers and end users

Testing process

Sample Test Manager Reports

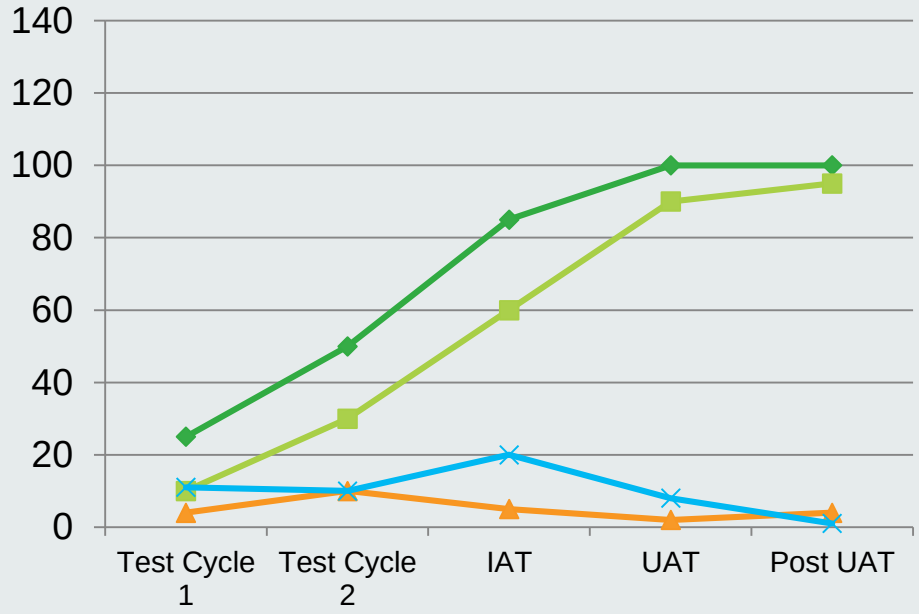
- Are you on or off track?

Scenario 1



- Defect rate does not decline
- Large number of active defects at UAT
- Acceptance unlikely

Scenario 2

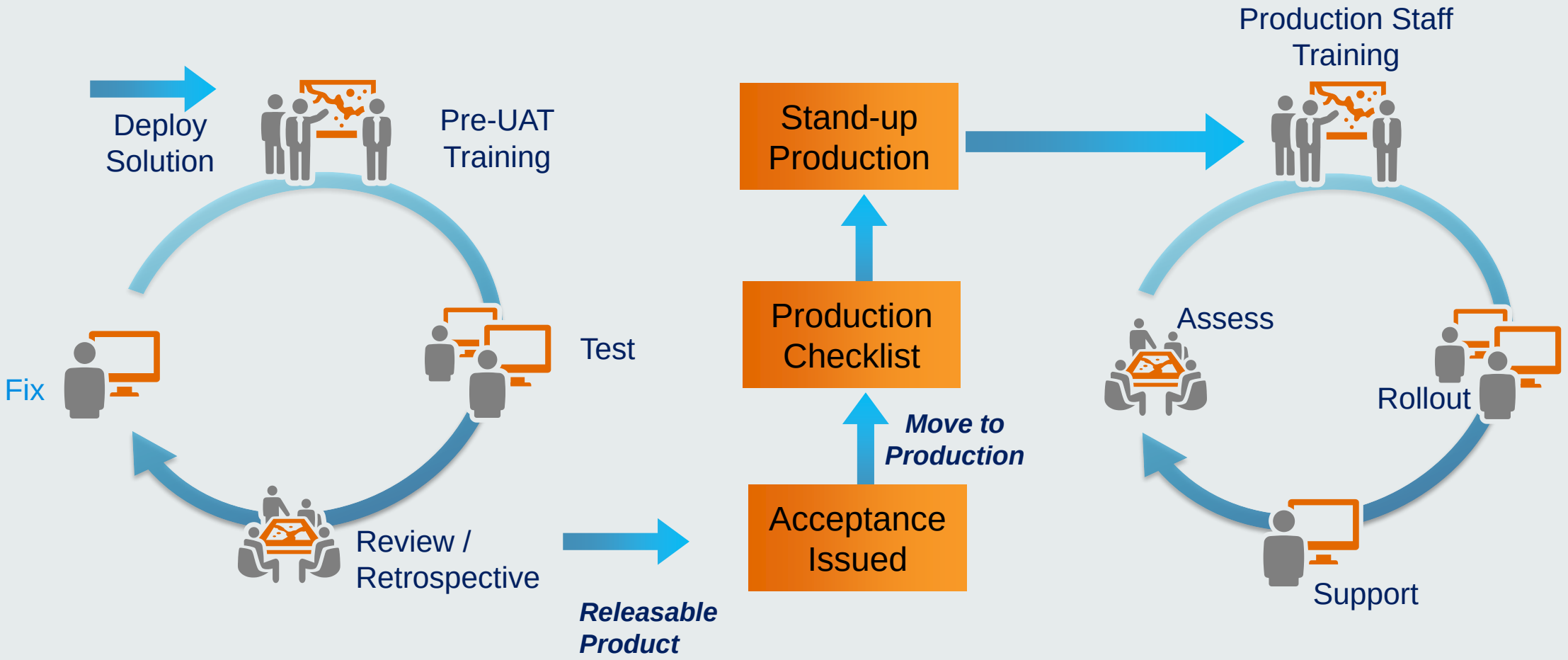


- Defect rate declining steadily over time
- Few active defects when entering UAT
- Acceptance likely

Deploy

Deploy

Typical Sequence of Events



Deploy

Plan Before you Go….

- Confirm the customer is ready
- Communicate all Hardware/Software pre-requisites
- Define resources, tasks, assignments
- Develop a detailed checklist
- Develop a detailed schedule

Appendix A
Afternoon
Page 46

Deployment Checklist

Deploy:

Typical Challenges

Multiple departments, locations	• Use an installation package • Plan sequencing with customer
Many system components (HW, RDBMS, GIS)	• Itemized list of components • Detailed checklist
Working around existing operational environment	• Separate environments • Move to operations in OFF hours
Varying skills in user community	• Targeted training program

Deploy

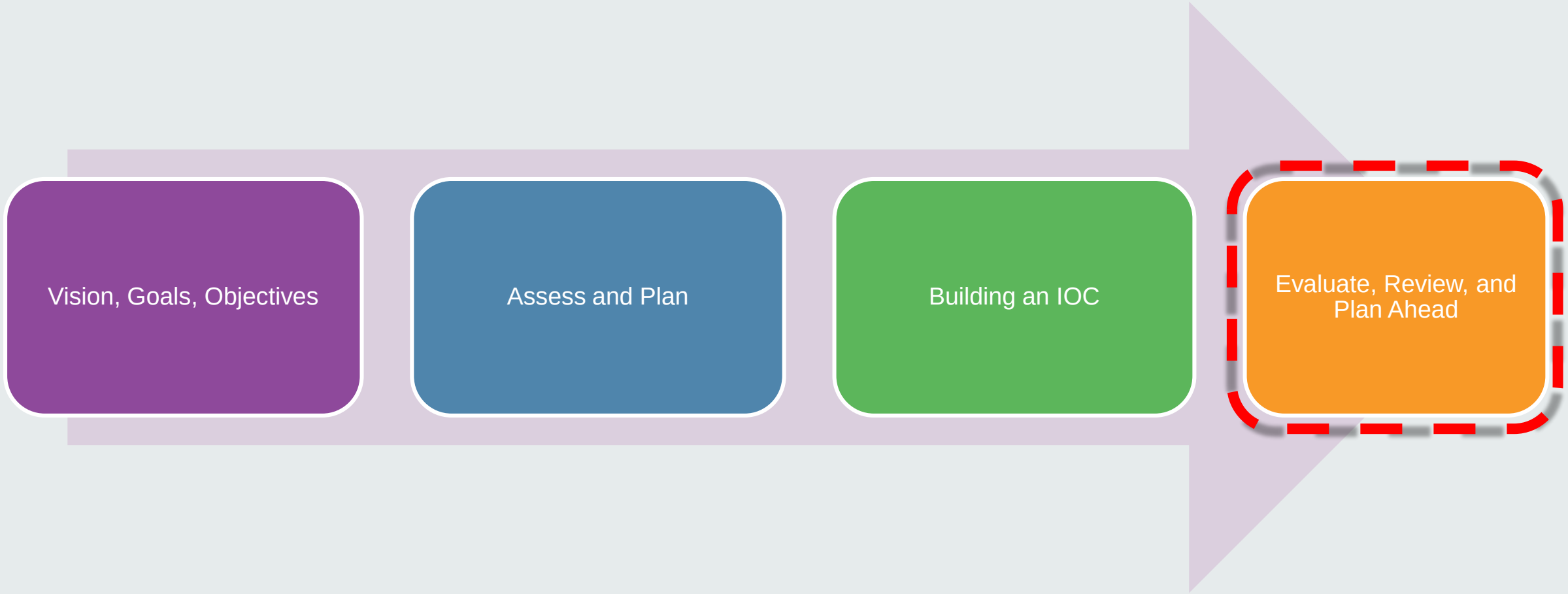
Tips and Techniques

- Swap staging and production
- Swap passive and active leg (for high availability)
- When upgrading, you must be able to fall back to previous configuration
- Down-time because of installation and upgrade still counts as down-time!

Training

- **Build training curriculum supporting GIS business processes**
  - **Targeted to the audience – not generic**
  - **Differentiate casual from advanced users**
  - **Help user visualize success in the production environment**
- **Sequencing and timing is essential**
  - **COTS before custom**
  - **Just-in-time is best**
- **Continue to grow GIS knowledge and capacity**

Assess the Following Components

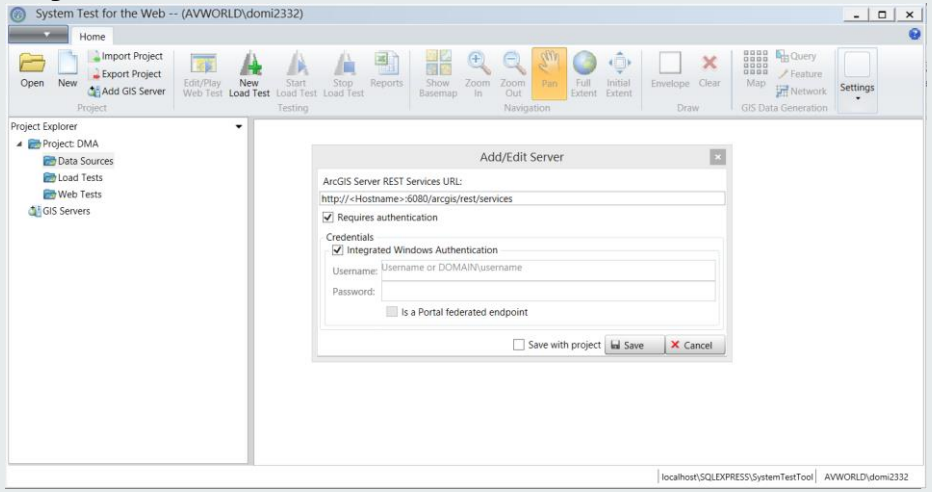


Evaluate

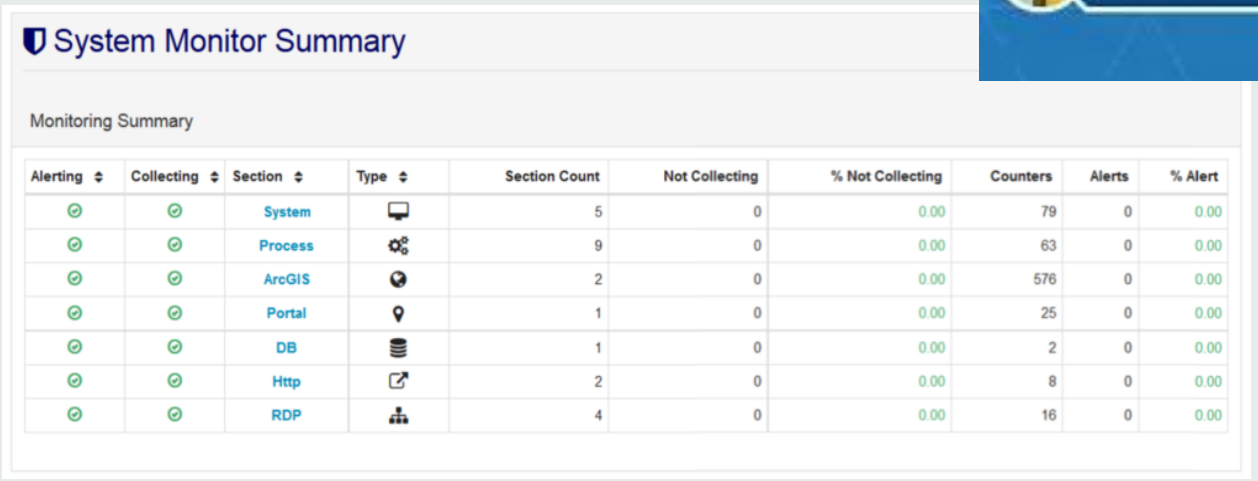
Performance Monitoring, Tuning

- **Monitor, evaluate**
  - **How is the system REALLY used?**
  - **Are there unanticipated bottlenecks?**
  - **Can performance be improved?**
- **Tuning can improve performance**
  - **RDBMS database, ArcGIS Server**
  - **Versioning workflows**

System Test



System Monitor

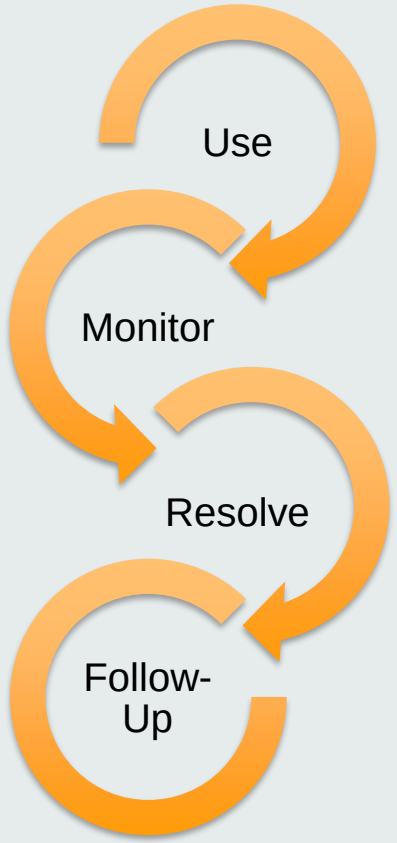


Alerting	Collecting	Section	Type	Section Count	Not Collecting	% Not Collecting	Counters	Alerts	% Alert
		System		5	0	0.00	79	0	0.00
		Process		9	0	0.00	63	0	0.00
		ArcGIS		2	0	0.00	576	0	0.00
		Portal		1	0	0.00	25	0	0.00
		DB		1	0	0.00	2	0	0.00
		Http		2	0	0.00	8	0	0.00
		RDP		4	0	0.00	16	0	0.00

Evaluate

Production Rollout / Transitioning / Operations

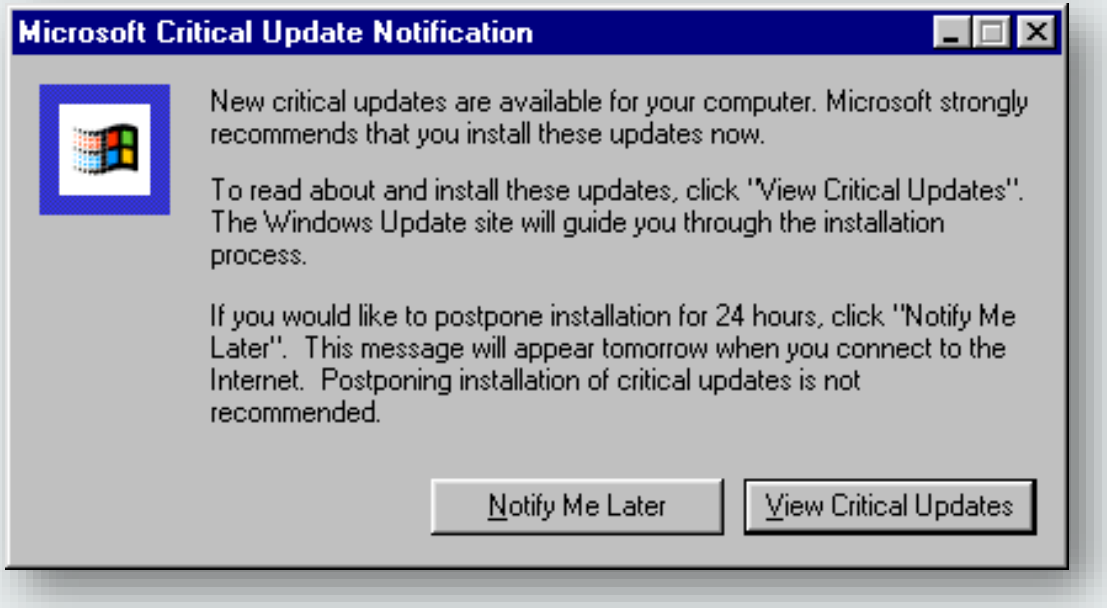
- **Users assume ownership**
- **A critical stage**
  - **Encourage use of the system**
  - **Avoid operational failures**
  - **Avoid abandonment of the system**
- **Plan to provide adequate support**
  - **Performance monitoring, tuning**
  - **Resolving issues (e.g. Help Desk)**
  - **Review and follow-up**



Evaluate

Version Updates

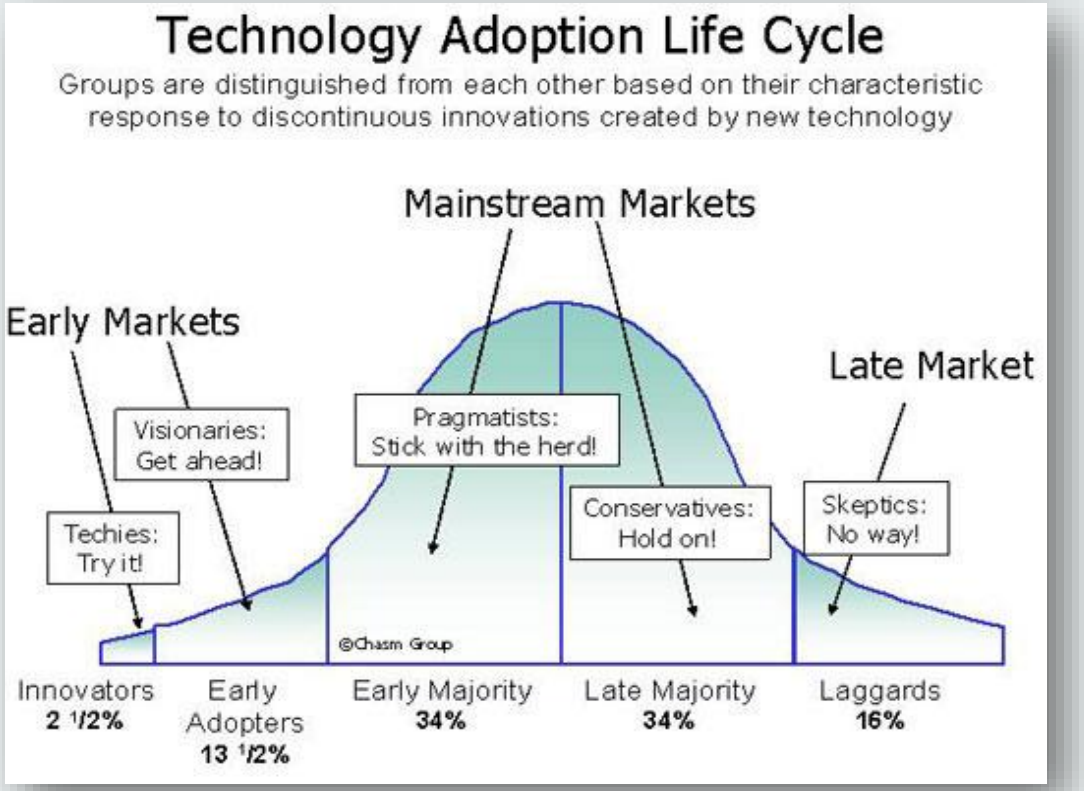
- **Evaluate options**
- **Upgrade in test environment first**
- **Includes Software Patches and OS Updates**
- **Communicate with IT**
- **Investigate new tools and functionality**
- **Get ready for rollback**
- **Is it worth it?**



Evaluate

Adopting Emerging Technologies

- **Risk/Reward**
- **Prototype new technology**
- **Demonstrate functionality to yourselves**
- **Demonstrate functionality to customer**
- **Consider Integration impacts**



Review

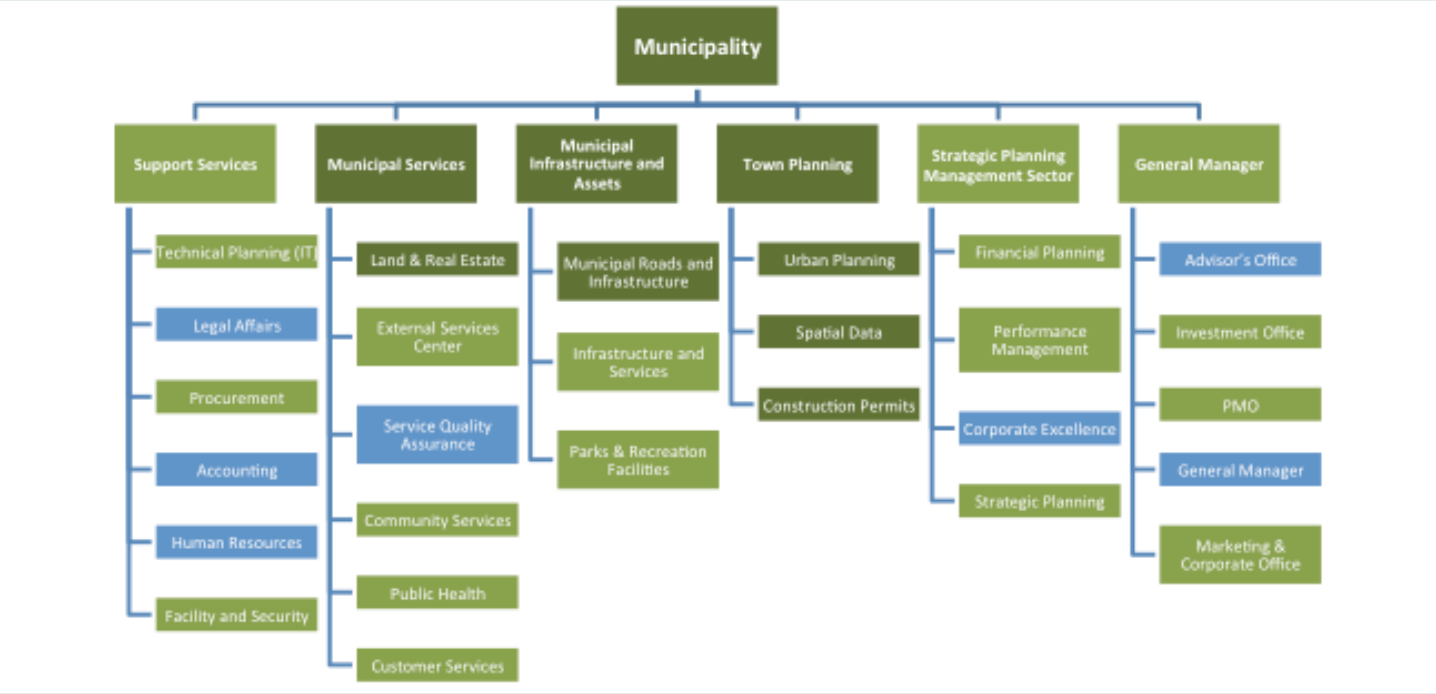
Measuring Benefits, Path Forward

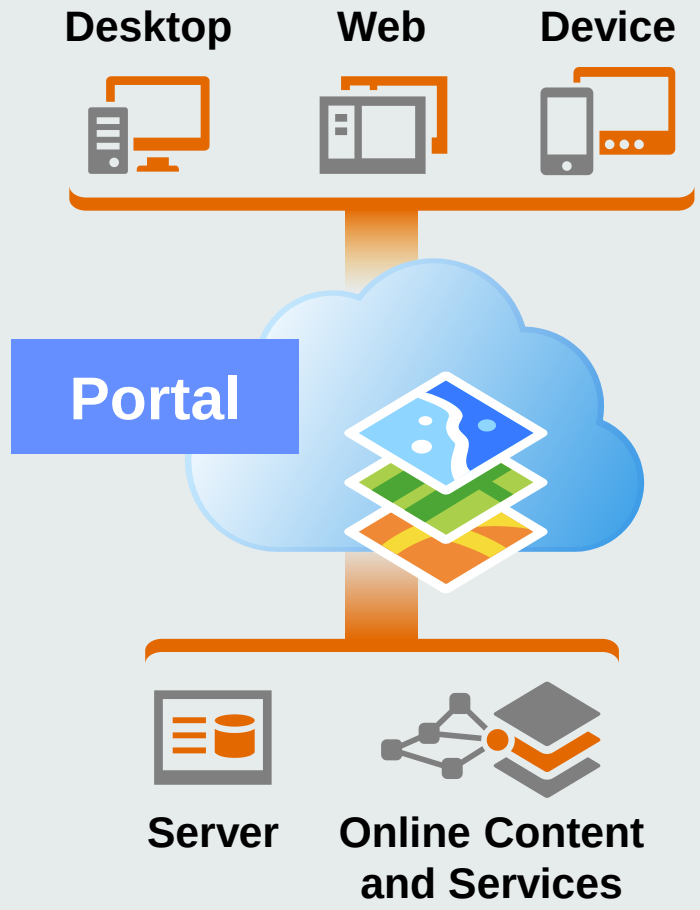
- **Operational review**
  - **Are existing expectations met?**
  - **What are future expectations?**
  - **Are there new requirements?**
  - **What benefits were realized?**
  - **Are there short comings?**
- **Assess the happiness of your stakeholders**
- **Make recommendations**
  - **Priorities for new data, services, apps**
  - **Organizational considerations**
  - **System tuning, upgrades**
- **Plan for Growth**

Plan Ahead

Increase organizational enablement

- New apps, data, services
- Further integration of organizational units



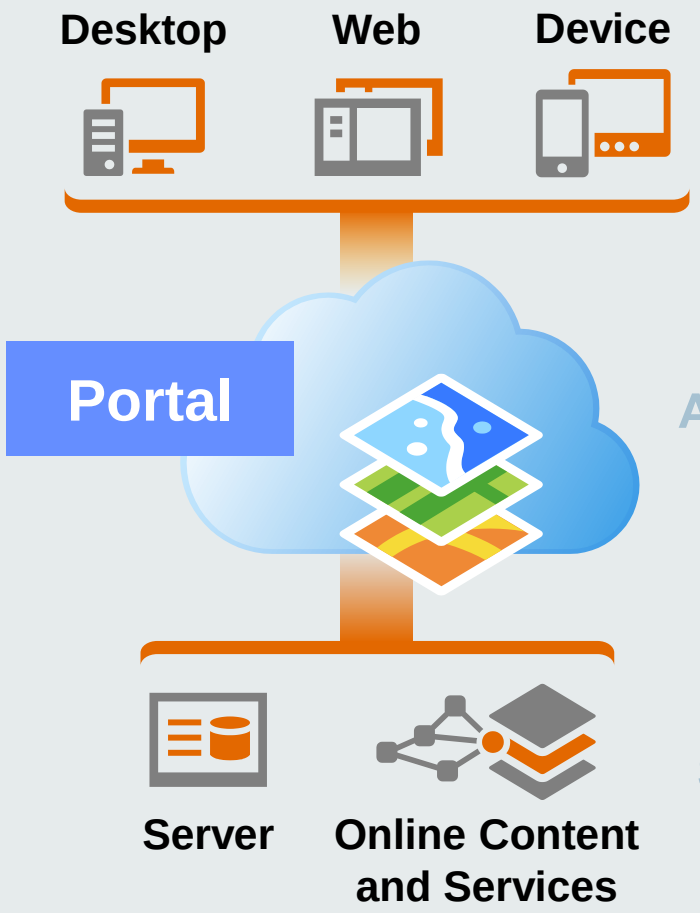


Moving to an Integrated Enterprise System

Initial Operating Capability

Foundation apps
Essential Information Products

“GIS Ready” data



Integrated Enterprise System

Apps
Extended / custom apps

Access & Identity
Integration with business systems

Advanced workflows

Services
Content production systems
Data modernization

Questions?

esri
THE
SCIENCE
OF
WHERE